



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN

Título del proyecto:

GESTIÓN DE SISTEMA DOMÓTICO MEDIANTE PLATAFORMA
WEB

Alumno: Carlos Ulibarrena Nuin

Tutor: José Javier Astrain Escola

Pamplona, 9 de Septiembre de 2011

INDICE

Capítulo 1. Introducción.....	3
1.1 Introducción.....	4
1.2 Objetivos.....	4
1.3 Definición del problema.....	4
1.4 Estado del arte.....	5
1.4 Solución propuesta.....	5
Capítulo 2. Análisis.....	7
2.1 Planificación.....	8
2.2 Metodología.....	10
2.3 Análisis.....	12
Capítulo 3. Diseño.....	68
3.1 Diseño del sistema.....	69
3.1.1 Arquitectura del sistema software....	69
3.1.2 Diseño de la capa de gestión de datos.....	70
3.1.3 Diseño de la capa de dominio.....	78
3.1.4 Diseño de la capa de presentación....	87
Capítulo 4. Conclusiones y líneas futuras.....	100
Capítulo 5. Bibliografía.....	103

Capítulo 1

Introducción

1.1 Introducción

En este proyecto se realiza la implementación de un sistema de control domótico para instalaciones agropecuarias. Mediante este sistema se pretende tener monitorizados una serie de elementos de las instalaciones para así poder observar posibles anomalías en el funcionamiento de los mismos. De esta manera, se podrá mantener en continua vigilancia a la instalación y se podrá modificar el funcionamiento de los elementos que la componen.

Este proyecto no deja de ser producto del continuo avance de las tecnologías y de la búsqueda de la mayor comodidad posible en la vida diaria del ser humano. Pero además, se debe añadir que con esta idea también se busca el incremento de la productividad de las instalaciones agropecuarias, ya que al conocer en todo momento el estado de la instalación, se pueden realizar acciones para la mejora del bienestar de los animales que se encuentran en la misma y se pueden reducir los consumos de las instalaciones.

1.2 Objetivos

El proyecto pretende desarrollar un nuevo modelo de centro de teleservicios que permita la gestión y control agrupado de explotaciones ganaderas intensivas de diferentes tipos, controlando de forma ininterrumpida las principales variables de las explotaciones, facilitando el trabajo deslocalizado de los gestores, la mejora de la productividad y el ahorro energético y de costes.

Para ello se va a realizar una aplicación web a la que tendrán acceso los diferentes usuarios que se encuentran comprometidos en el proyecto (instaladores, centro de teleservicios y granjeros). De esta forma, cada usuario podrá acceder únicamente a los servicios del sistema que le sean necesarios. Así un granjero no podrá modificar los parámetros de configuración de los sensores que tiene instalados en su granja, ya que estas configuraciones sólo las puede llevar a cabo el instalador.

1.4 Definición del problema

El problema que queremos tratar es cómo conseguir un sistema que sea capaz de manejar una serie de variables existentes en un aparato domótico situado en cualquier punto del mundo al que tenemos acceso mediante una conexión de red.

Las variables que vamos a manejar contienen los valores de funcionamiento de los aparatos domóticos que se encuentran instalados en la granja. Por lo tanto, estos valores nos indican el estado en que se encuentra cada aparato para así poder mostrarlo en el sistema, o si el aparato lo permite, para poder modificar los valores de funcionamiento del aparato.

Mediante este sistema deberemos de ser capaces de controlar los elementos domóticos que se encuentren conectados en una granja y así poder realizar cambios en el funcionamiento de los mismos, o realizar una monitorización de sus valores. Por ejemplo, encender una bombilla o realizar una gráfica con las distintas temperaturas que se están dando en la granja.

1.4 Estado del arte

Sistemas de control de explotaciones ganaderas ya existen, pero son habitualmente individuales y manejados in-situ, por lo que no cumplen con uno de nuestros objetivos principales, que es facilitar el trabajo deslocalizado de los gestores de las instalaciones. Además, no existe ningún sistema de control de explotaciones ganaderas que funcione con un sistema domótico específicamente creado para ello, sino que simplemente se han llegado a utilizar algunos componentes de la domótica doméstica para comprobar sus posibles aplicaciones en el ámbito de las instalaciones agropecuarias.

En cuanto a centros de Telecontrol o de Teleservicios también existen pero aplicados a otros sectores (p.ej. energético. transporte...), por lo que se observa la necesidad de crear un centro de Telecontrol específico para las necesidades que queremos abordar.

La novedad de este proyecto estriba en los siguientes aspectos:

- Aplicación e integración de la domótica en las explotaciones (soluciones hasta ahora aplicadas principalmente en el ámbito doméstico o industrial).
- Desarrollo de un modelo de centro de teleservicios adaptado a la realidad de la gestión de las explotaciones agrarias.
- Aplicación de soluciones de movilidad, para compatibilizar centralización (manejo centralizado de datos y prestación de servicios compartidos) y disponibilidad (acceso por los gestores de las explotaciones a los datos y a la gestión en cualquier momento y lugar).

1.5 Solución propuesta

Ante lo expuesto anteriormente se ha considerado la siguiente solución como la mejor para llevar a cabo el proyecto:

Se deberá crear un centro de control que nos muestre el estado real de cada granja del sistema. Con esta aplicación se podrán detectar anomalías en el funcionamiento de los sistemas instalados en las granjas y así poder actuar para corregir dichas anomalías.

Se deberá realizar una aplicación mediante la cual deberemos ser capaces de controlar las variables que se encuentren instaladas en las distintas instalaciones agropecuarias del sistema, para así poder realizar cambios en los estados de las mismas (ej. subir una persiana, encender un ventilador). Esta aplicación será accesible para los gerentes de cada instalación (lógicamente cada gerente podrá acceder sólo a sus instalaciones) y también será accesible para el personal del centro de control, ya que ante una anomalía

tendrá que poder modificar las variables necesarias para que la instalación vuelva a funcionar correctamente.

Además, se tendrá que realizar una aplicación para los técnicos que instalen el sistema domótico en las instalaciones agropecuarias. Mediante esta aplicación, especificarán los parámetros de configuración de los distintos elementos que se han instalado.

Por último, se deberá crear una aplicación para el mantenimiento de los diferentes componentes del sistema: usuarios, instalaciones agropecuarias, etc.

La aplicación debe de ser realizada con la tecnología Microsoft Silverlight 4, una estructura para la aplicaciones web que agrega nuevas funciones multimedia como la reproducción de videos, gráficos vectoriales, animaciones e interactividad, en forma similar a lo que hace Adobe Flash.

El sistema requiere la utilización de una arquitectura cliente-servidor, en la que tendremos una serie de clientes web que se van a conectar a un servidor web desde donde obtendrán los datos de sus granjas. Estos datos se deberán de almacenar en una base de datos se deberán de almacenar en una base de datos para mantenerlos de forma consistente y evitar la pérdida de los mismos.

Para tener actualizados los datos del sistema, más concretamente los datos de funcionamiento de las granjas, se deberán de realizan refrescos periódicos de las diferentes variables domóticas que están instaladas en las granjas. De este modo, cada vez que un usuario quiera consultar el estado de una granja, podrá obtener los datos más recientes de la misma desde la base de datos ubicada en el servidor.

Capítulo 2

Análisis

En este capítulo se realiza el análisis de la solución propuesta, mostrando la planificación temporal que se debe realizar para conseguir la finalización del proyecto, así como la elección de la metodología que se va a utilizar para el desarrollo del mismo. También se realiza un análisis de los diferentes casos de uso que se dan en el sistema y se especifica las características de cada uno.

2.1 Planificación

En esta sección se presentan las actividades que se deben de realizar para completar satisfactoriamente el proyecto. La Imagen 1 ilustra el diagrama Gantt que describe dichas actividades, y que está basado en los partes de horas de la empresa en la que estuve de prácticas mientras realizaba el proyecto.

Entre las actividades que se deben de realizar, cabe destacar la gran cantidad de horas que se deben de emplear en la investigación de la tecnología utilizada para realizar el proyecto, así como la investigación de nuevos controles de usuario para poder mostrar los valores de las variables de las instalaciones domóticas y para poder controlarlas.

También, hay que reseñar la formación necesaria para aprender las técnicas de programación en Silverlight 4, ya que esta forma de programación introduce nuevas técnicas para la obtención de los datos desde las bases de datos, así como nuevas formas de estructurar las aplicaciones. Además, no existen muchos ejemplos complejos de programaciones con Silverlight 4, por lo que los ejemplos a tomar como referencia son escasos y muchas veces insuficientes. Debido a esta escasez de documentación sobre técnicas avanzadas en la programación con Silverlight 4, cuando se conseguía obtener una nueva documentación generalmente había que realizar ajustes en la programación ya realizada.

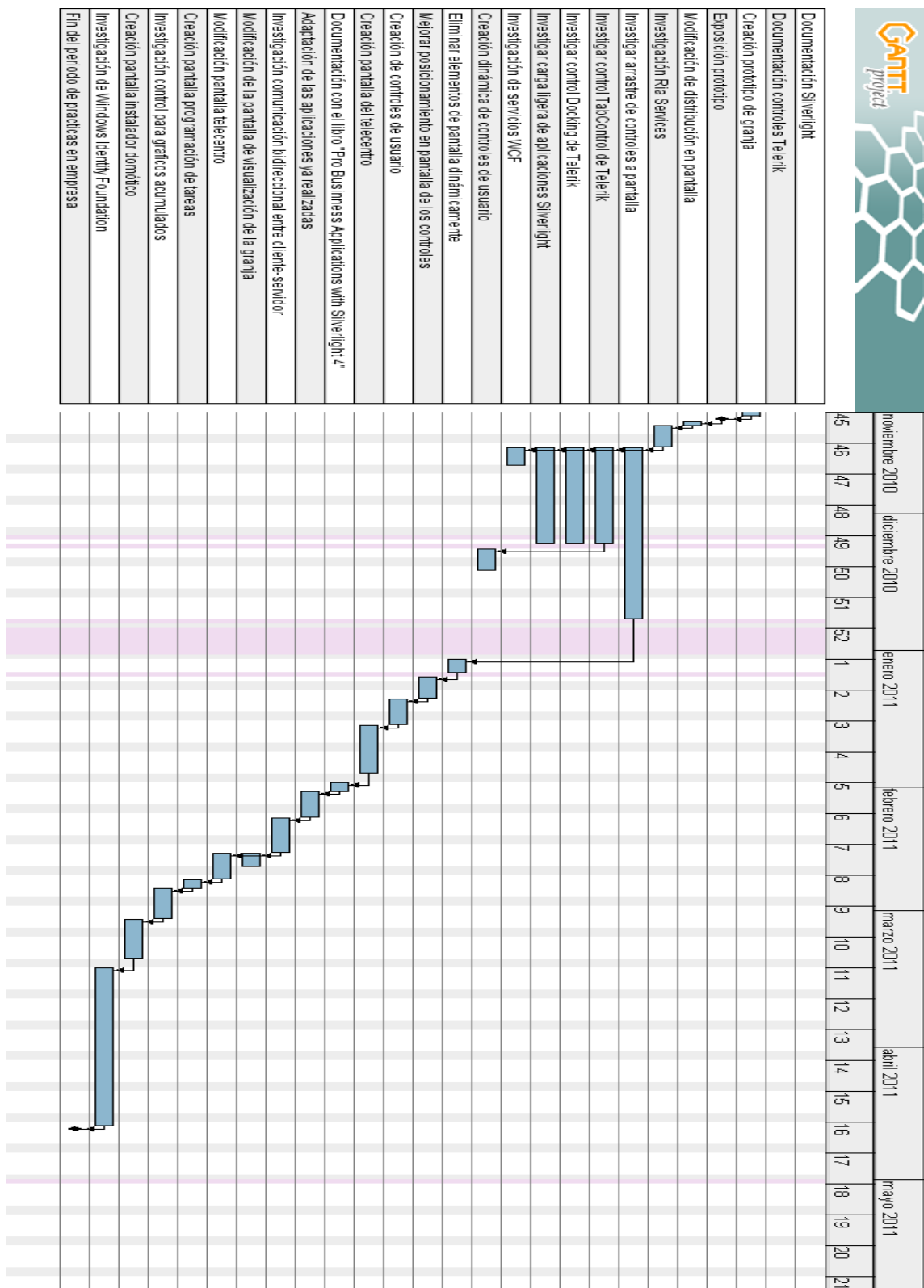


Imagen 1: Diagrama de Gantt del proyecto.

2.2 Metodología

Antes de profundizar en el desarrollo de este proyecto, me gustaría concretar lo que es una metodología. Una metodología es el conjunto de pasos, procedimientos, técnicas, herramientas y el soporte documental que ayudan a desarrollar un producto software.

La orientación a objetos es un paradigma de la ingeniería del software que basa la arquitectura de los sistemas en los datos/objetos que el sistema manipula. Estos objetos representan las entidades físicas y conceptuales del mundo real relacionadas con la aplicación.

En la evolución de la orientación a objetos hubo una avalancha de notaciones y técnicas que se solucionaron con la estandarización, que dio lugar al método conocido con el nombre de Método Unificado. Este posteriormente se transformó en UML (Lenguaje de Modelado Unificado) que es el que usaremos en este proyecto.

2.2.1. ¿Por qué usar orientación a objetos?

El desarrollo de un sistema consiste en una serie de iteraciones del tipo división-unión; hay que dividir el problema para comprenderlo e ir uniendo las partes divididas para construir la solución.

El proceso de división se realiza separando las distintas funciones del sistema y descomponiendo estas en subfunciones sucesivamente hasta obtener elementos simples. La arquitectura de un programa así realizado es un reflejo del sistema estudiado. Este método aporta resultados satisfactorios cuando las funciones están bien definidas y son estables en el tiempo. Sin embargo, al existir un fuerte acoplamiento entre arquitectura y funciones, las evoluciones importantes pueden suponer modificaciones estructurales importantes.

La descomposición orientada a objetos se basa en los objetos que integran la estructura y el comportamiento. Las funciones se representan mediante colaboraciones entre los objetos que componen el sistema.

El empleo de las metodologías orientadas a objetos se justifica con las siguientes razones:

- La orientación a objetos se acerca más a la forma de pensar de las personas, haciéndolo más comprensible y fácil de aplicar.
- Facilita la abstracción, centrándonos en cada momento únicamente en los elementos que nos interesen descartando los demás.
- Aumenta la consistencia interna al tratar los atributos y las operaciones como un todo.
- Permite expresar características comunes sin repetir la información.
- Facilita la reutilización de diseños y código.
- Facilita la revisión y modificación de sistemas desarrollados.
- Origina sistemas más estables y robustos.

2.2.2 Modelo de proceso

El desarrollo de un producto software comercial supone un gran esfuerzo que puede durar desde varios meses a un año o más. Lo ideal es dividir el trabajo en pequeñas partes. Cada parte es una iteración que da lugar a un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo y los incrementos al crecimiento del producto.

En cada iteración los desarrolladores identifican y especifican los casos de uso relevantes; los casos de uso nos proporcionan un diseño utilizando la arquitectura seleccionada como guía, implementan el sistema mediante componentes y verifican que los componentes satisfacen dichos casos de uso.

Las iteraciones construyen los modelos resultantes incremento por incremento, pasando cada iteración por el estudio de requisitos, análisis, diseño, implementación y pruebas.

2.2.3 Modelo de proceso basado en componentes

Por tanto, después de observar las características del proyecto y evaluando las diferentes posibilidades me he decantado por utilizar la metodología de Modelo de proceso basado en componentes. El modelo de proceso a utilizar es el basado en componentes. Dicho modelo es evolutivo por naturaleza y exige un enfoque iterativo.

Un componente es una pieza de código preelaborado que encapsula alguna funcionalidad expuesta a través de interfaces estándar. Es algo muy similar a lo que podemos observar en el equipo de música que tenemos en nuestra sala. Cada componente de aquel aparato ha sido diseñado para acoplarse perfectamente con sus pares, las conexiones son estándar y el protocolo de comunicación está ya preestablecido. Al unirse las partes, obtenemos música para nuestros oídos. [1]

El proceso orientado a objetos se mueve a través de una espiral evolutiva que comienza con la comunicación del usuario. Es aquí donde se define el dominio del problema y se identifican las clases básicas del sistema. La planificación y el análisis de riesgos establecen una base para el plan del proyecto orientado a objetos.

La ingeniería del software hace hincapié en la reutilización, por tanto, debemos buscar las clases en una biblioteca de clases orientadas a objetos antes de construir una. Cuando una clase no existe en la biblioteca, el desarrollo de software aplica análisis, diseño, programación y pruebas para crear la clase y los objetos derivados de la misma. La nueva clase se pone en la biblioteca de manera que pueda ser utilizada en cualquier momento.

He optado por este modelo de proceso porque se adapta perfectamente a las características del sistema ya que permite la reutilización del software, simplifica las pruebas, simplifica el mantenimiento del sistema y se obtiene una mayor calidad, dado que un componente puede ser construido y luego mejorado continuamente por un experto u organización.

2.3 Análisis

En esta sección voy a realizar el análisis de las especificaciones que debe de cumplir el proyecto, según una serie de valoraciones obtenidas en las reuniones con los expertos en materia de instalaciones agropecuarias y domótica. A partir de este análisis se realizará el desarrollo de la aplicación, por lo que es una parte fundamental para el buen desarrollo del proyecto.

2.3.1 Modelo de casos de uso

Un caso de uso especifica el comportamiento deseado del sistema, representa sus requisitos funcionales. Describe un conjunto de secuencias de interacciones o escenarios: un flujo principal y flujos alternativos o excepcionales. Se deben de describir todos los escenarios posibles.

A continuación se describen los casos de uso siguiendo la plantilla de D. Coleman y los diagramas de secuencia genéricos. Los he dividido por el actor que interactúa con los casos de uso, por lo que tendremos 3 diagramas diferentes y habrá algún caso de uso que pueda estar en más de un diagrama. En ese caso, la descripción del caso de uso aparecerá sólo una vez.

Diagrama de casos de uso para el usuario Granjero

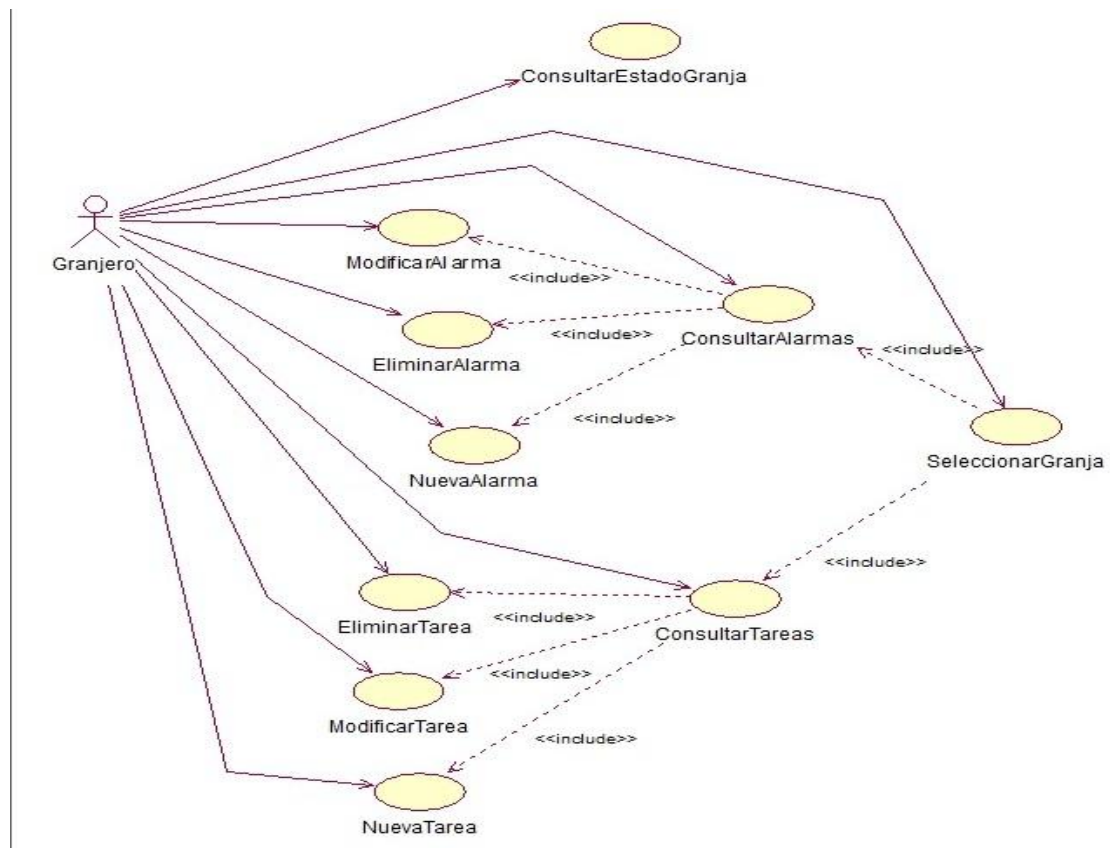


Imagen 2: Diagrama de casos de uso para el usuario Granjero.

En la Imagen 2 podemos observar los diferentes casos de uso que se han analizado en el sistema para el usuario Granjero. Estas son todas las acciones que puede llevar a cabo este usuario según el análisis que se ha realizado del sistema. A continuación se explica cada uno de los casos de uso por separado, mostrando además los diagramas de secuencia y de estados que tienen asociados.

- **Caso de uso: ConsultarEstadoGranja**

1. Descripción: Para cada granja del sistema, se debe de tener controlado en todo momento su estado, es decir, se debe de comprobar que todos los controles de usuario que hay instalados en la granja funcionan con normalidad y que no producen ninguna alarma.
2. Actores: AdministradorTelecentro, Granjero
3. Precondiciones:
4. Pasos:
5. Variaciones:
6. Requisitos no funcionales: esto va a ser la parte principal de una zona de control que se va a encargar de controlar todas las granjas del sistema, por lo que se debe de visualizar de manera rápida y sencilla si una granja está en estado de alarma.
7. Cuestiones:
8. Diagrama de secuencia:

1. AdministradorTelecentro

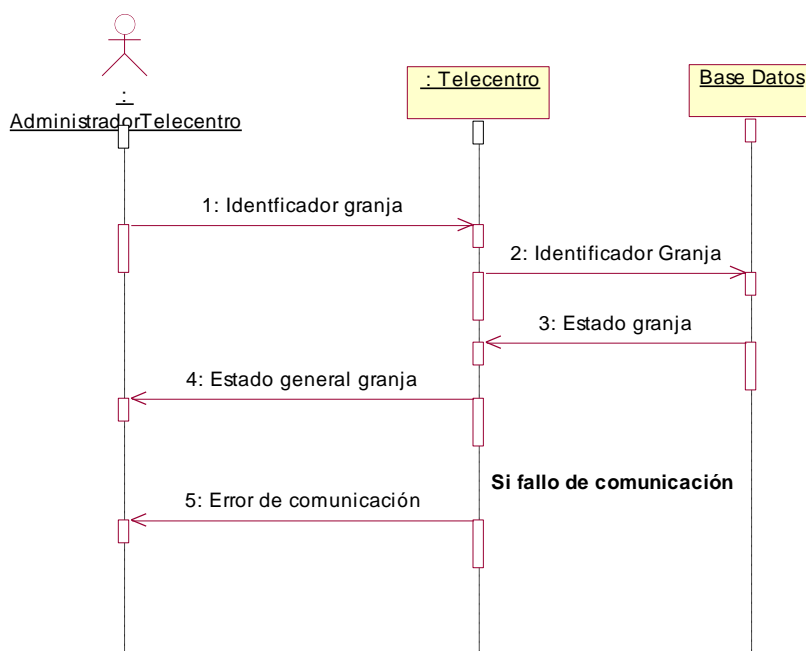


Imagen 3: Diagrama de secuencia del usuario AdministradorTelecentro para el caso de uso ConsultarEstadoGranja.

2. Granjero

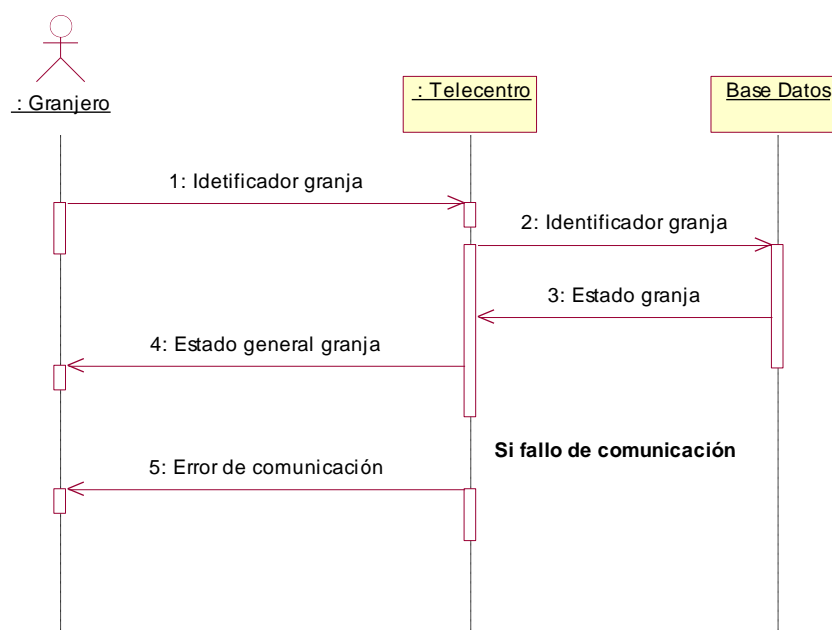


Imagen 4: Diagrama de secuencia del usuario Granjero para el caso de uso ConsultarEstadoGranja.

9. Diagrama de estados:

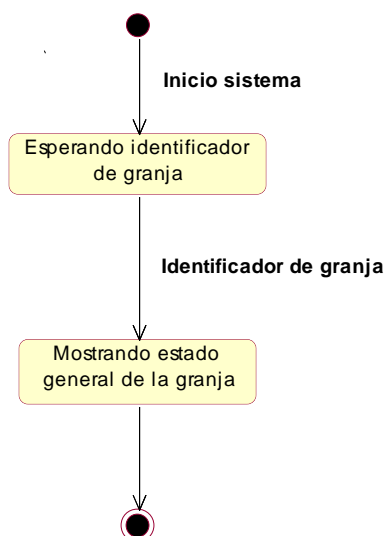


Imagen 5: Diagrama de estados para el caso de uso ConsultarEstadoGranja.

- **Caso de uso: ConsultarAlarmas**

1. Descripción: Para un control de usuario que pertenece a una granja, se debe de poder consultar las alarmas que tiene programadas (si es que el control de usuario contiene algún registro de salida, ya que las alarmas se programan para los registros de salida de un control de usuario).
2. Actores: AdministradorTelecentro, Granjero
3. Precondiciones:
 1. Se debe seleccionar previamente una granja dada de alta en el sistema.
4. Pasos:
 1. Seleccionar el control de usuario del que queremos consultar las alarmas.
 2. Pulsar el botón para mostrar las alarmas del control de usuario.
 3. Seleccionar el registro sobre el que se han programado las alarmas (puede haber varios).
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

1. AdministradorTelecentro

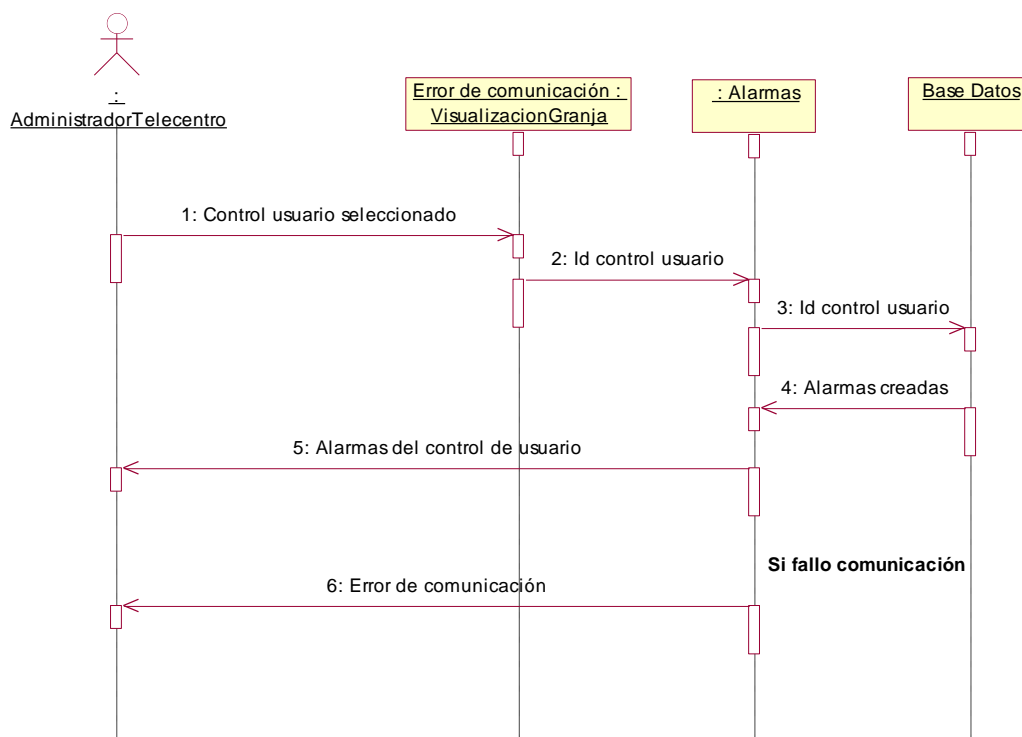


Imagen 6: Diagrama de secuencia del usuario AdministradorTelecentro para el caso de uso ConsultarAlarmas.

2. Granjero

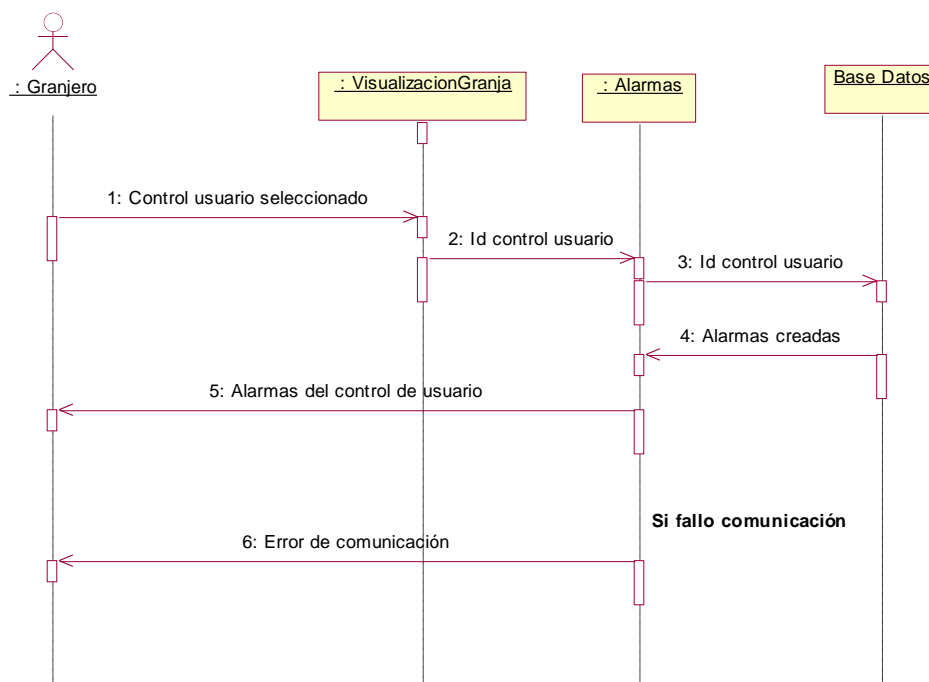


Imagen 7: Diagrama de secuencia del usuario Granjero para el caso de uso ConsultarAlarmas.

9. Diagrama de estados:

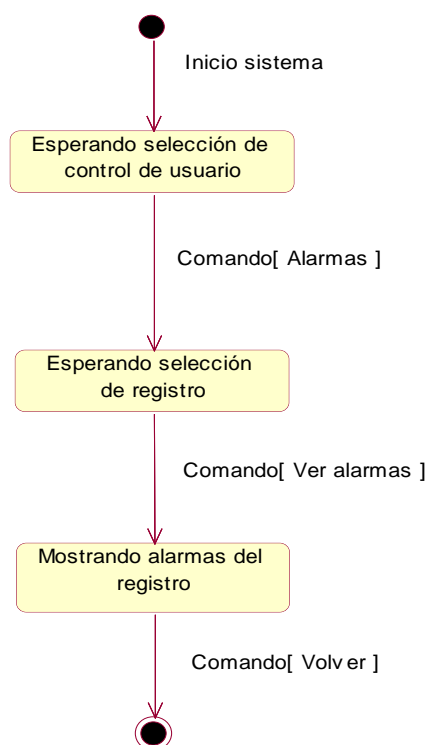


Imagen 8: Diagrama de estados para el caso de uso ConsultarAlarmas.

- **Caso de uso: ModificarAlarma**

1. Descripción: Para un control de usuario que pertenece a una granja, se debe de poder modificar las alarmas que tenga programadas.
2. Actores: AdministradorTelecentro, Granjero
3. Precondiciones:
 1. Se deben de consultar previamente las alarmas que tiene el control de usuario.
4. Pasos:
 1. Seleccionar la alarma que se quiere modificar.
 2. Introducir los datos que se van a modificar.
 3. Guardar los datos.
 4. Si son correctos se mostrará un mensaje.
 5. Si no son correctos se mostrará un mensaje y se señalará cuales son incorrectos.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

1. AdministradorTelecentro

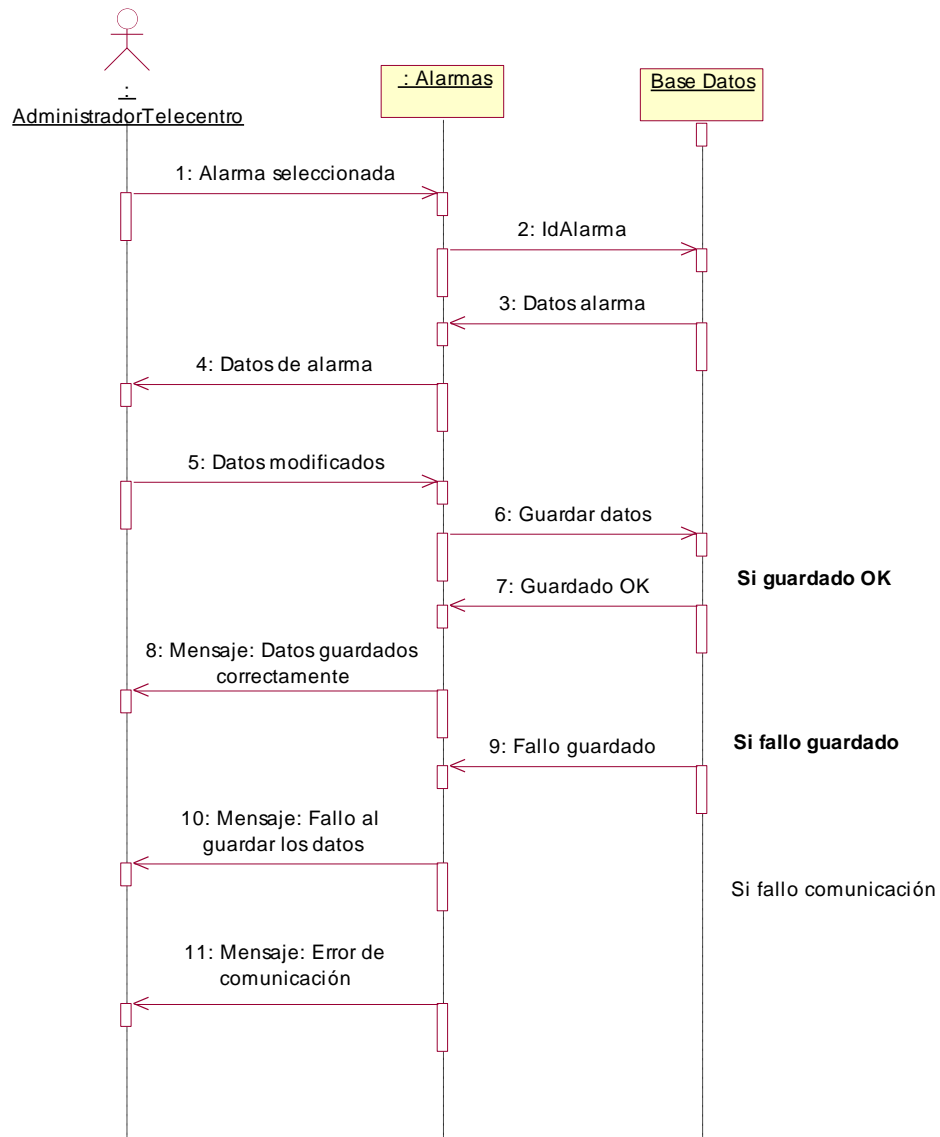


Imagen 9: Diagrama de secuencia del usuario AdministradorTelecentro para el caso de uso ModificarAlarma.

2. Granjero

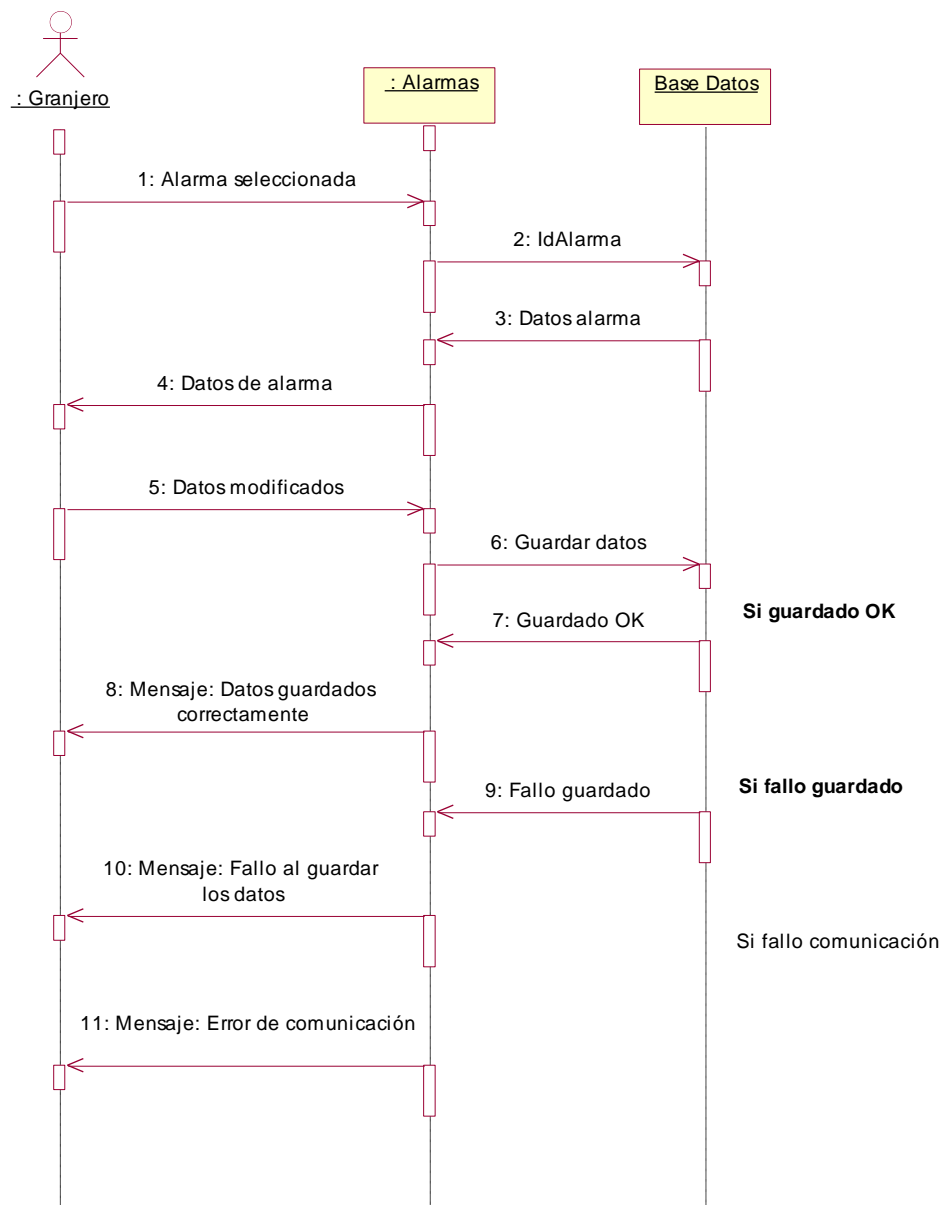


Imagen 10: Diagrama de secuencia del usuario Granjero para el caso de uso ModificarAlarma.

9. Diagrama de estados:

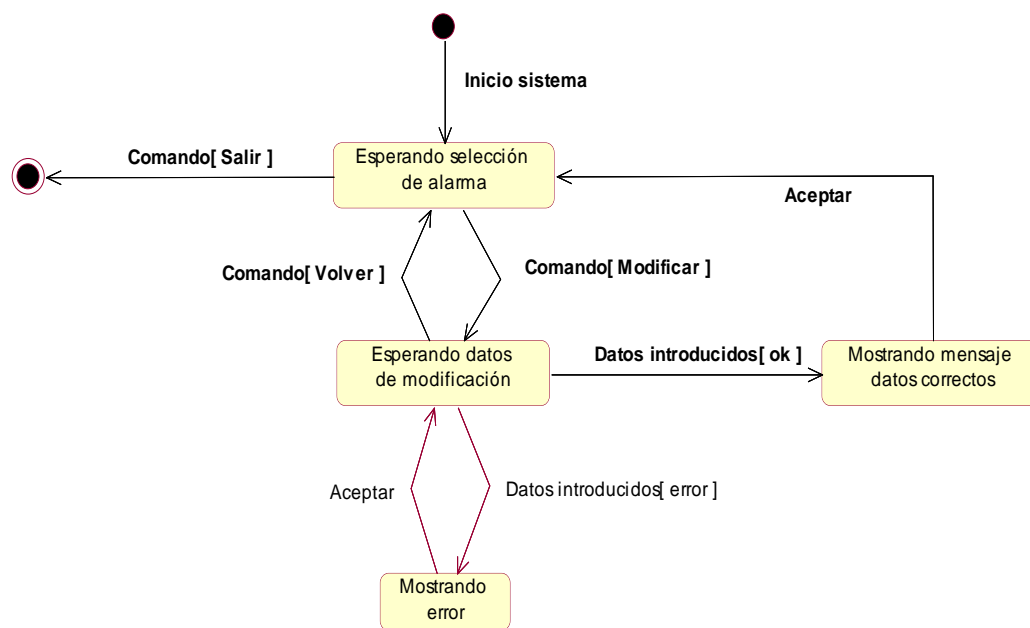


Imagen 11: Diagrama de estados para el caso de uso ModificarAlarma.

- **Caso de uso: EliminarAlarma**

1. Descripción: Para un control de usuario que pertenece a una granja, se debe de poder eliminar alarmas existentes.
2. Actores: AdministradorTelecentro, Granjero
3. Precondiciones:
 1. Se deben de consultar previamente las alarmas que tiene el control de usuario.
4. Pasos:
 1. Seleccionar la alarma que se quiere eliminar.
 2. Eliminarla.
 3. Mostrar mensaje si se ha eliminado correctamente.
 4. Mostrar mensaje si ha dado fallo la eliminación.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

1. AdministradorTelecentro

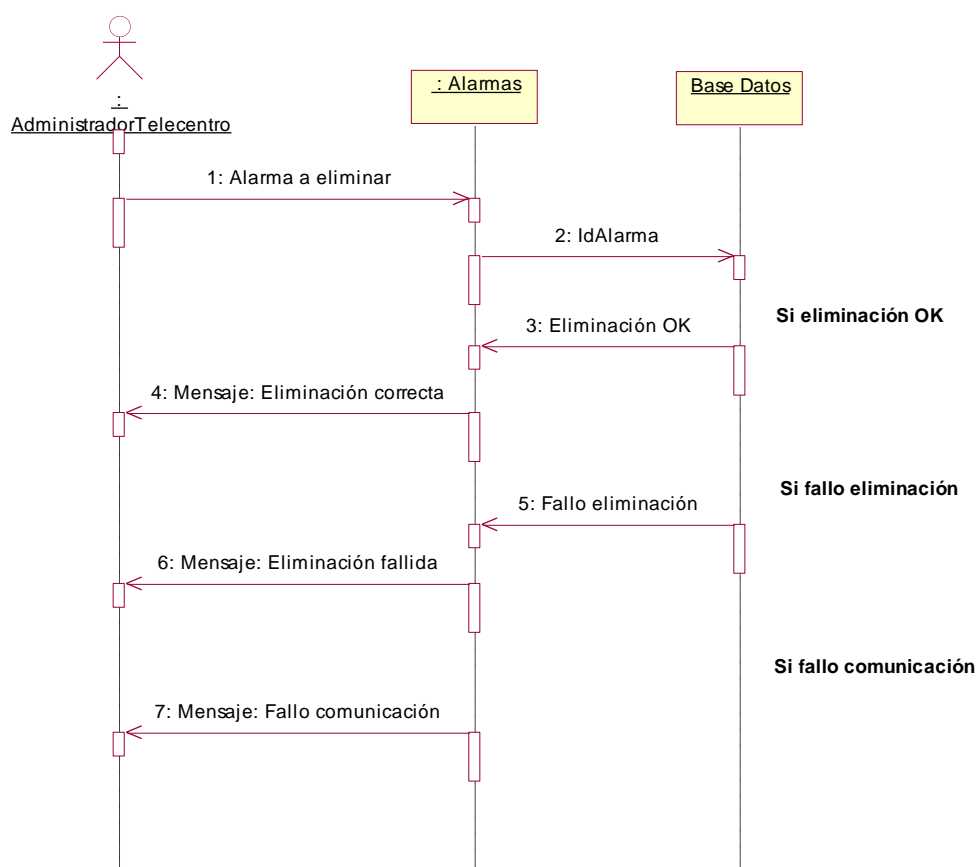


Imagen 12: Diagrama de secuencia del usuario AdministradorTelecentro para el caso de uso EliminarAlarma.

2. Granjero

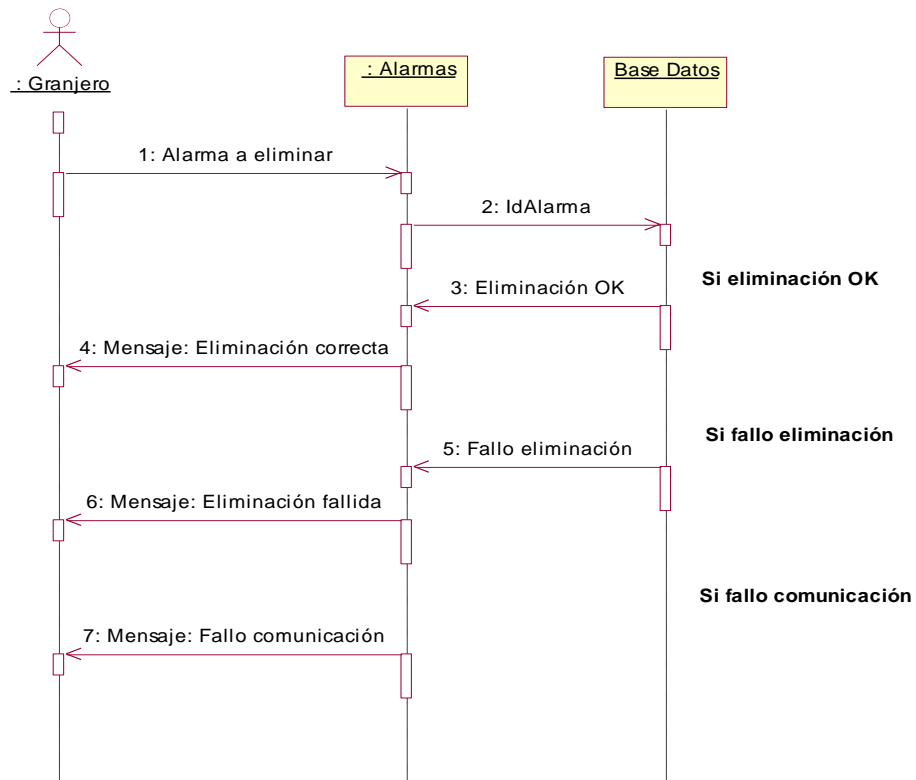


Imagen 13: Diagrama de secuencia del usuario Granjero para el caso de uso EliminarAlarma.

9. Diagrama de estados:

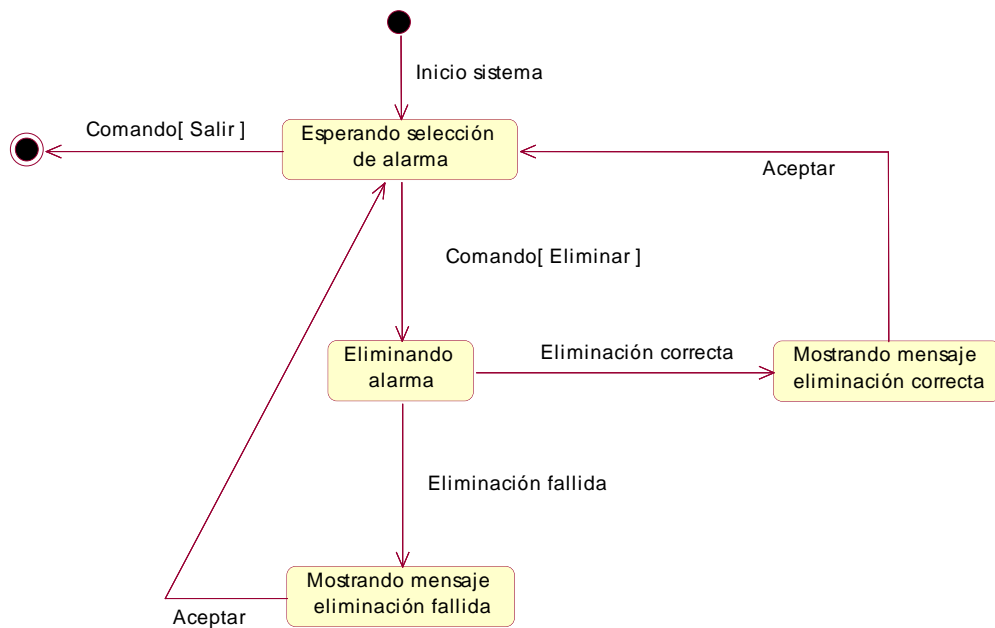


Imagen 14: Diagrama de estados para el caso de uso EliminarAlarma.

- **Caso de uso: NuevaAlarma**

1. Descripción: Para un control de usuario que pertenece a una granja, se debe de poder agregar nuevas alarmas.
2. Actores: AdministradorTelecentro, Granjero
3. Precondiciones:
 1. Se deben de consultar previamente las alarmas que tiene el control de usuario para asegurarse de que ya no existe una igual.
4. Pasos:
 1. Seleccionar que se va a introducir una nueva alarma.
 2. Introducir los datos de la nueva alarma.
 3. Si los datos son correctos, mostrar un mensaje y guardarlos.
 4. Si los datos son incorrectos, mostrar un mensaje e indicar cuales han dado fallo.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

1. AdministradorTelecentro

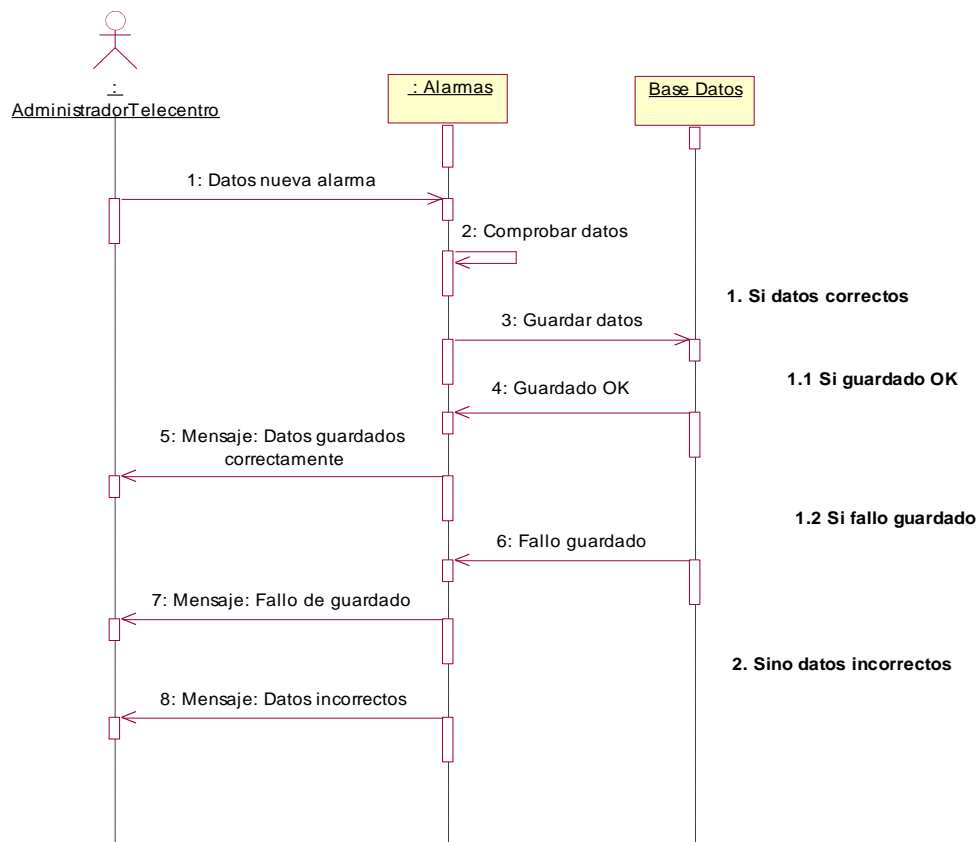


Imagen 15: Diagrama de secuencia del usuario AdministradorTelecentro para el caso de uso NuevaAlarma.

2. Granjero

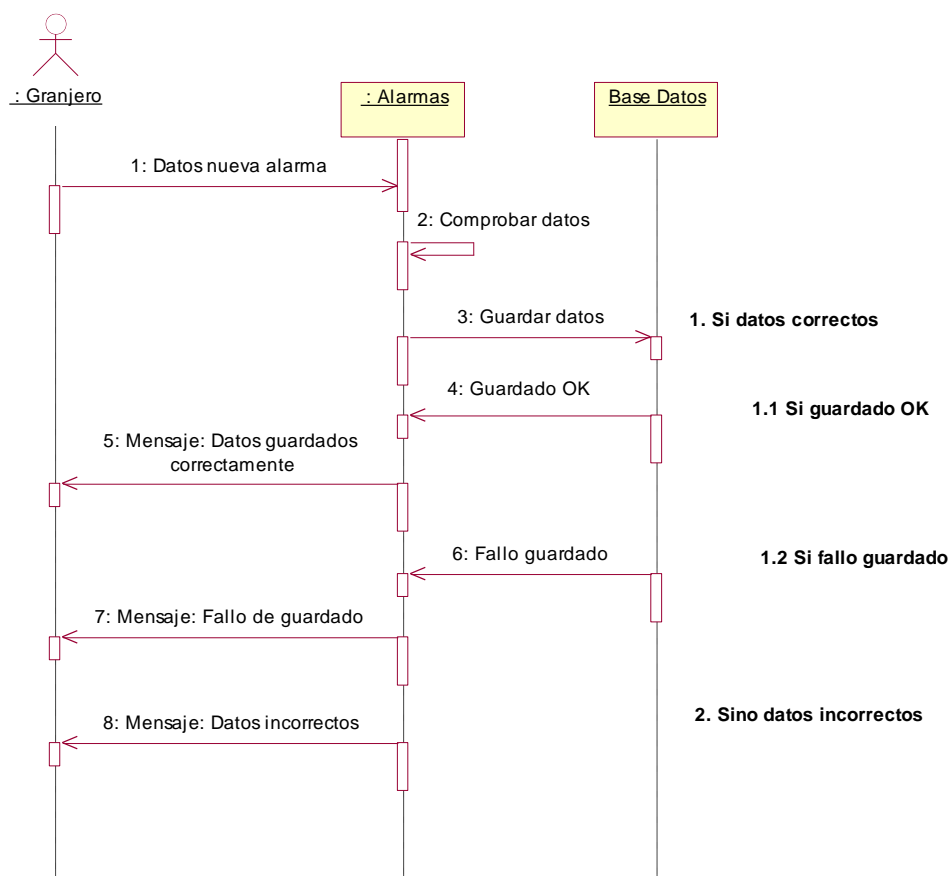


Imagen 16: Diagrama de secuencia del usuario Granjero para el caso de uso NuevaAlarma.

9. Diagrama de estados:

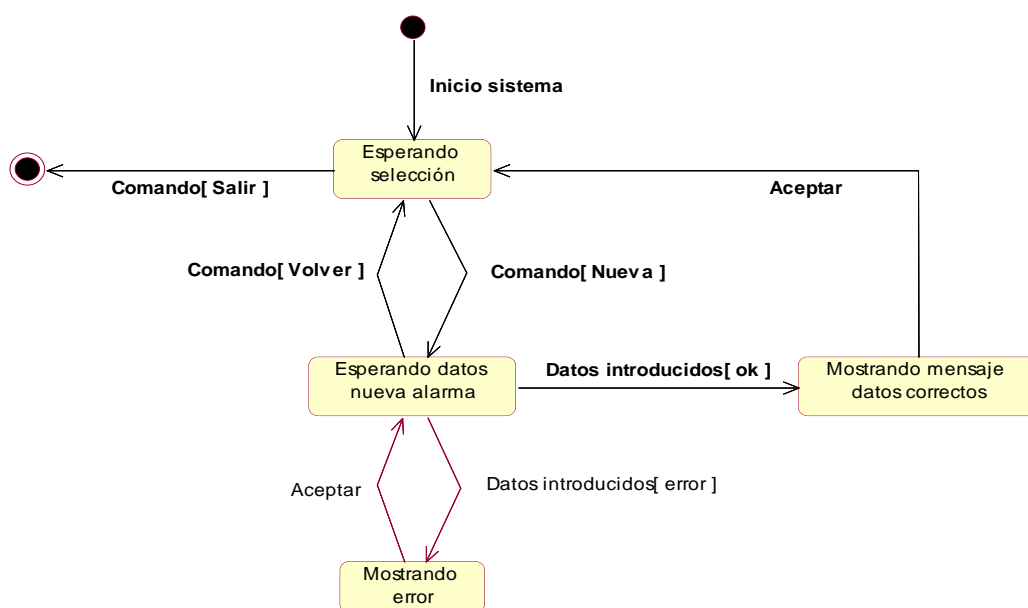


Imagen 17: Diagrama de estados para el caso de uso NuevaAlarma.

- **Caso de uso: ConsultarTareas**

1. Descripción: Para un control de usuario que pertenece a una granja, se debe de poder consultar las tareas que tiene programadas.
2. Actores: AdministradorTelecentro, Granjero
3. Precondiciones:
 1. Se debe seleccionar previamente una granja dada de alta en el sistema.
4. Pasos:
 1. Seleccionar el control de usuario del que queremos consultar las tareas.
 2. Pulsar el botón para mostrar las tareas del control de usuario.
 3. Seleccionar el registro sobre el que se han programado las tareas (puede haber varios).
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

1. AdministradorTelecentro

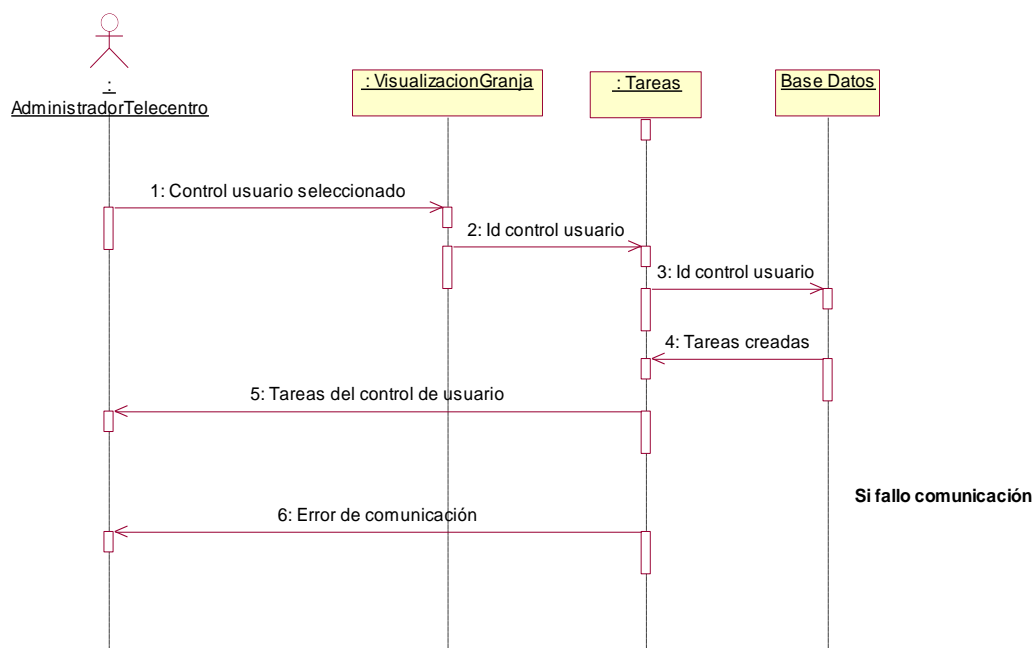


Imagen 18: Diagrama de secuencia del usuario AdministradorTelecentro para el caso de uso ConsultarTareas.

2. Granjero

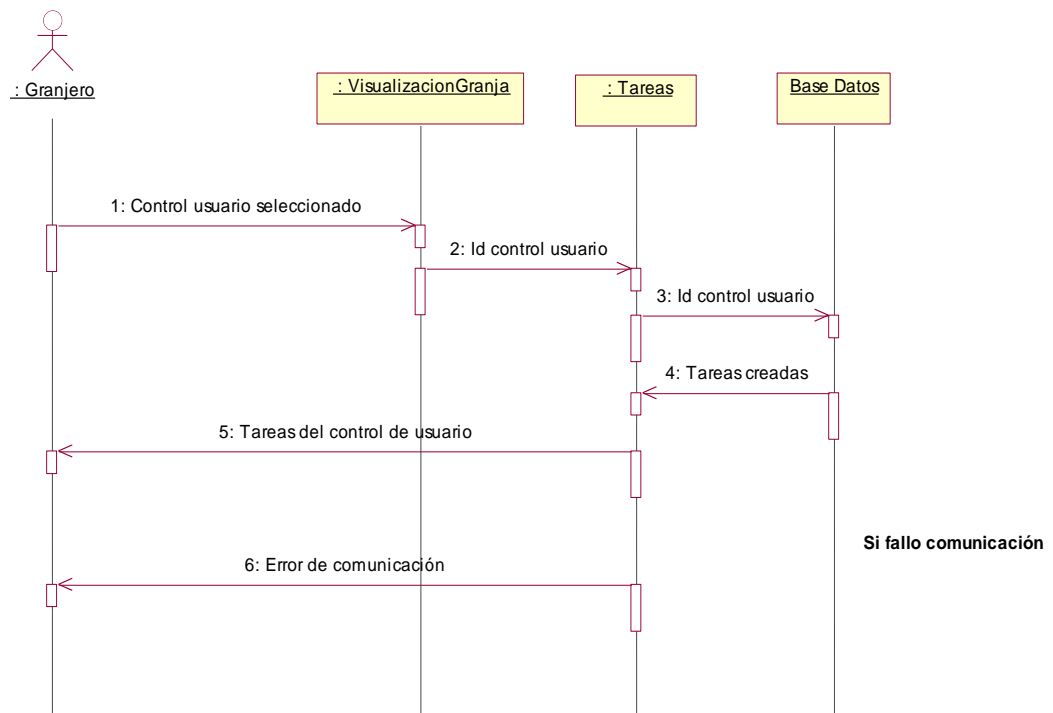


Imagen 19: Diagrama de secuencia del usuario Granjero para el caso de uso ConsultarTareas.

9. Diagrama de estados:

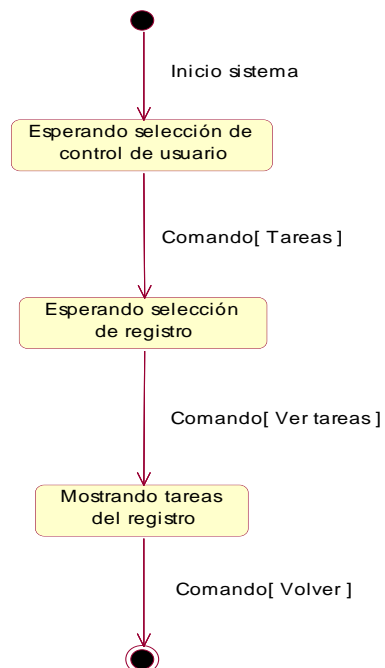


Imagen 20: Diagrama de estados para el caso de uso ConsultarTareas.

- **Caso de uso: EliminarTarea**

1. Descripción: Para un control de usuario que pertenece a una granja, se debe de poder eliminar tareas existentes.
2. Actores: AdministradorTelecentro, Granjero
3. Precondiciones:
 1. Se deben de consultar previamente las tareas que tiene el control de usuario.
4. Pasos:
 1. Seleccionar la tarea que se quiere eliminar.
 2. Eliminarla.
 3. Mostrar mensaje si se ha eliminado correctamente.
 4. Mostrar mensaje si ha dado fallo la eliminación.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

1. AdministradorTelecentro

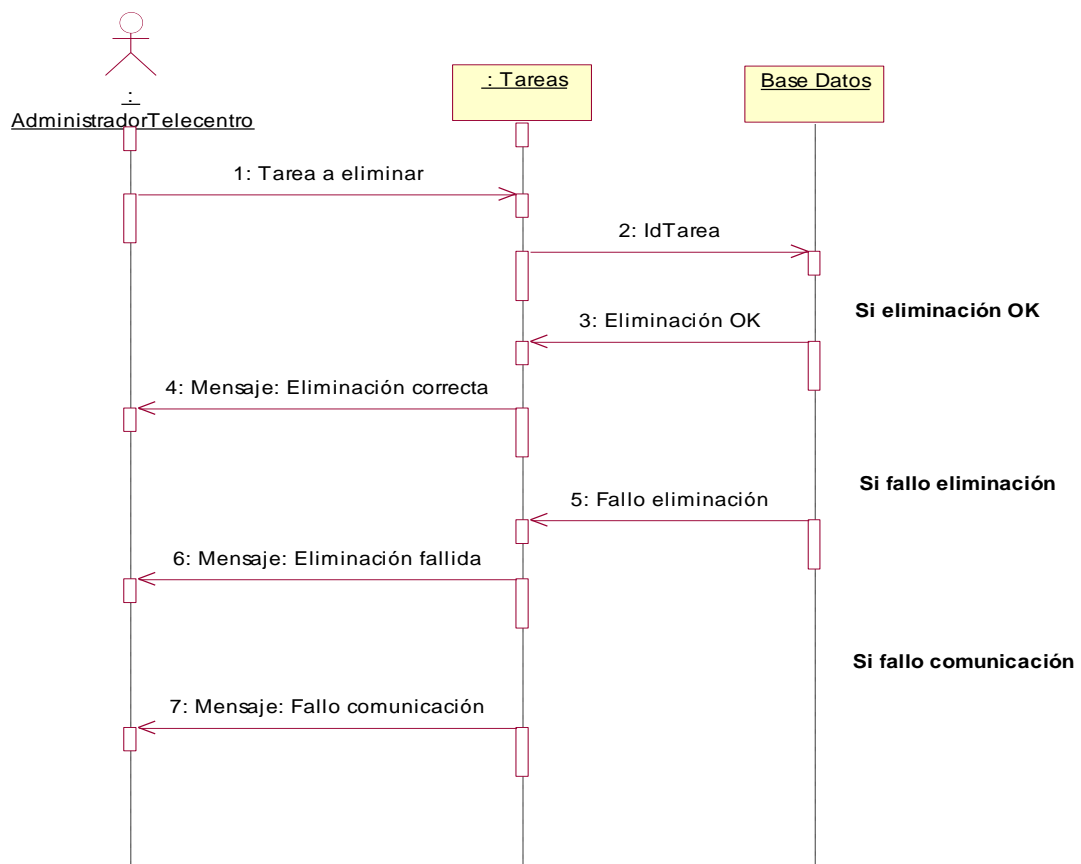


Imagen 21: Diagrama de secuencia del usuario AdministradorTelecentro para el caso de uso EliminarTarea.

2. Granjero

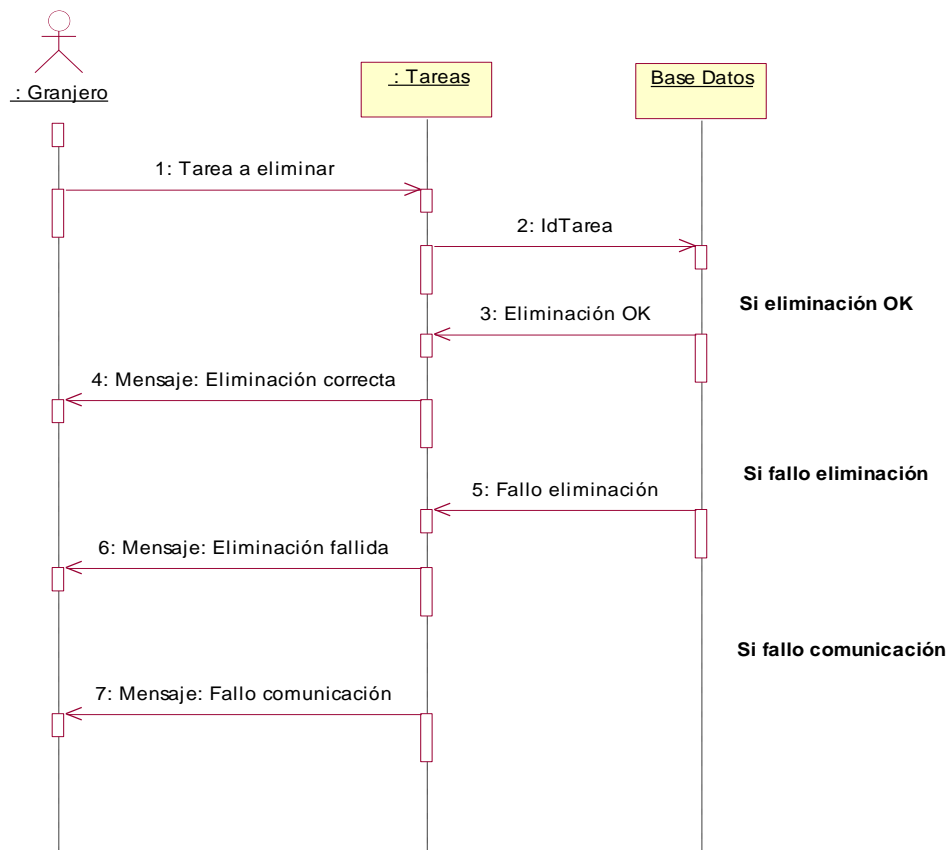


Imagen 22: Diagrama de secuencia del usuario Granjero para el caso de uso EliminarTarea.

9. Diagrama de estados:

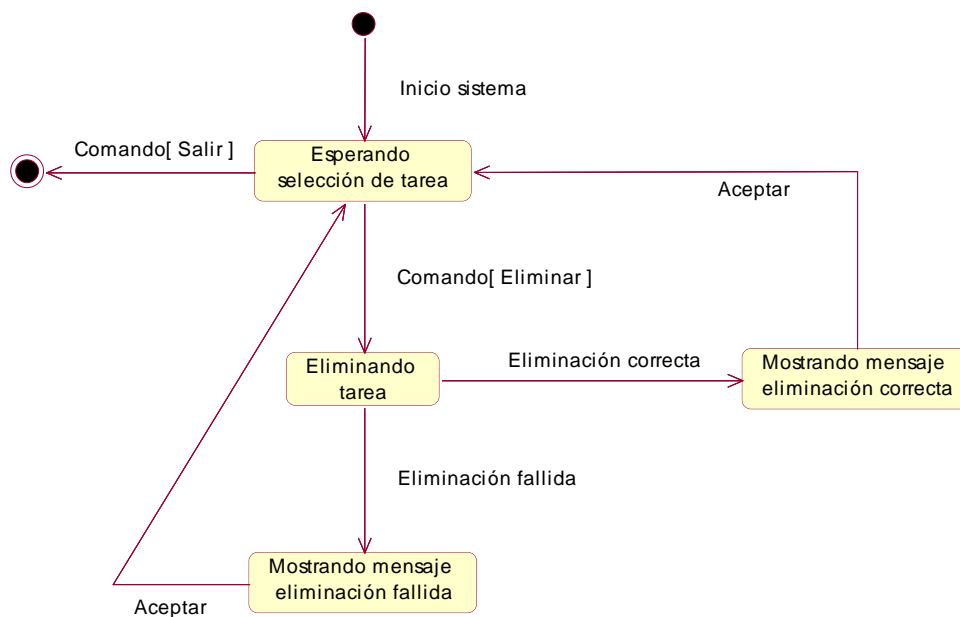


Imagen 23: Diagrama de estados para el caso de uso EliminarTarea.

- **Caso de uso: ModificarTarea**

1. Descripción: Para un control de usuario que pertenece a una granja, se debe de poder modificar las tareas que tenga programadas.
2. Actores: AdministradorTelecentro, Granjero
3. Precondiciones:
 1. Se deben de consultar previamente las tareas que tiene el control de usuario.
4. Pasos:
 1. Seleccionar la tarea que se quiere modificar.
 2. Introducir los datos que se van a modificar.
 3. Guardar los datos.
 4. Si son correctos se mostrará un mensaje.
 5. Si no son correctos se mostrará un mensaje y se señalará cuales son incorrectos.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

1. AdministradorTelecentro

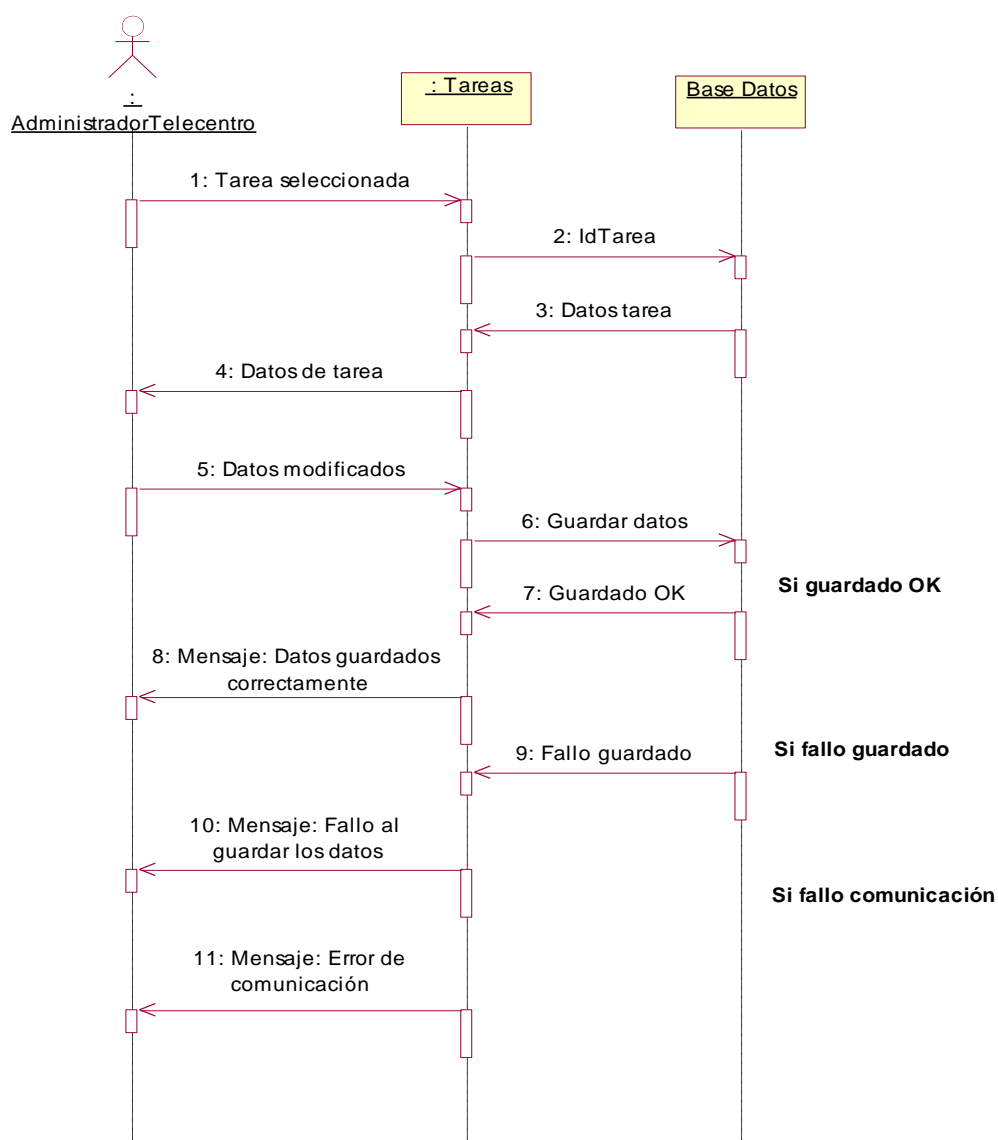


Imagen 24: Diagrama de secuencia del usuario AdministradorTelecentro para el caso de uso ModificarTarea.

2. Granjero

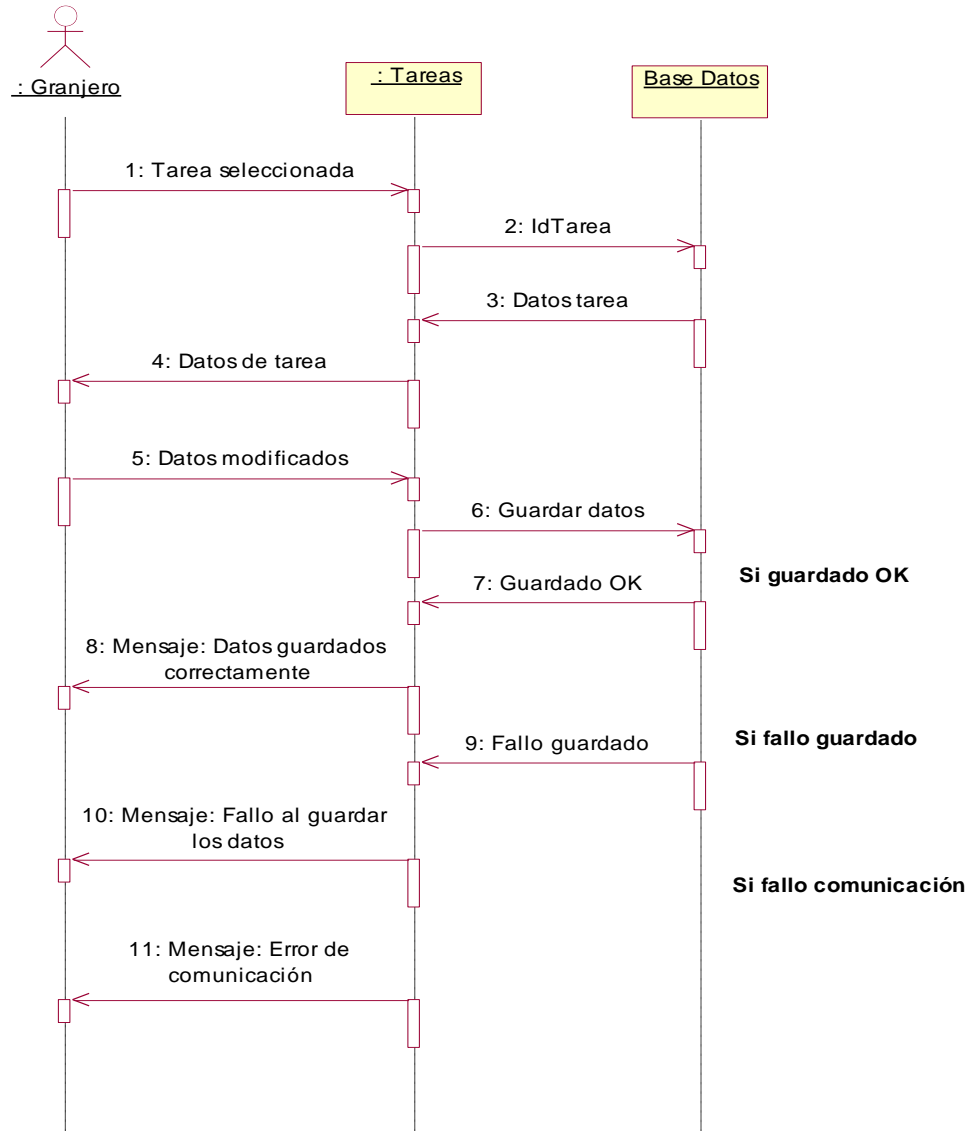


Imagen 25: Diagrama de secuencia del usuario Granjero para el caso de uso ModificarTarea.

9. Diagrama de estados:

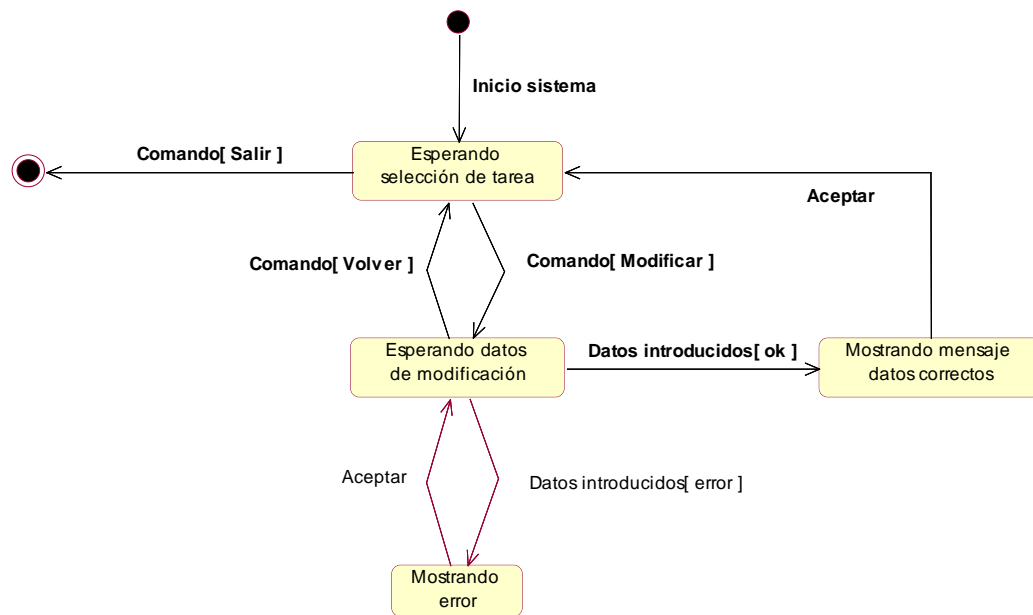


Imagen 26: Diagrama de estados para el caso de uso ModificarTarea.

- **Caso de uso: NuevaTarea**

1. Descripción: Para un control de usuario que pertenece a una granja, se debe de poder agregar nuevas tareas.
2. Actores: AdministradorTelecentro, Granjero
3. Precondiciones:
 1. Se deben de consultar previamente las tareas que tiene el control de usuario para asegurarse de que ya no existe una igual.
4. Pasos:
 1. Seleccionar que se va a introducir una nueva tarea.
 2. Introducir los datos de la nueva tarea.
 3. Si los datos son correctos, mostrar un mensaje y guardarlos.
 4. Si los datos son incorrectos, mostrar un mensaje e indicar cuales han dado fallo.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

1. AdministradorTelecentro

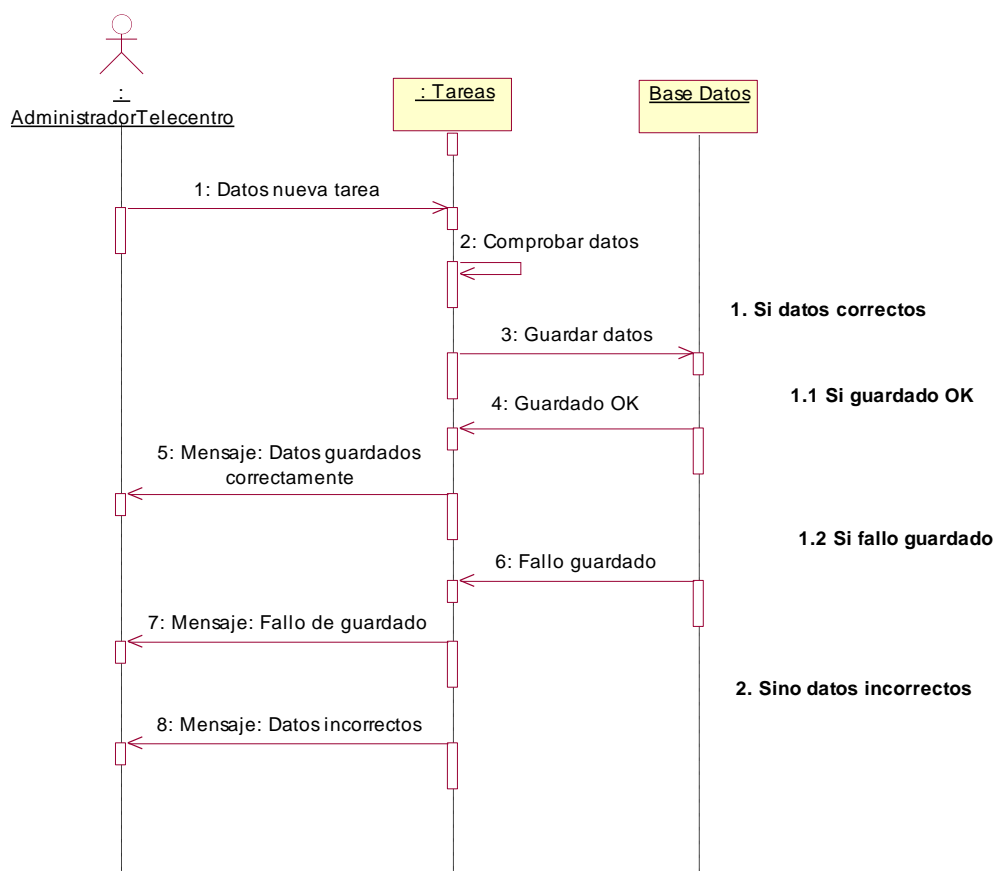


Imagen 27: Diagrama de secuencia del usuario AdministradorTelecentro para el caso de uso NuevaTarea.

2. Granjero

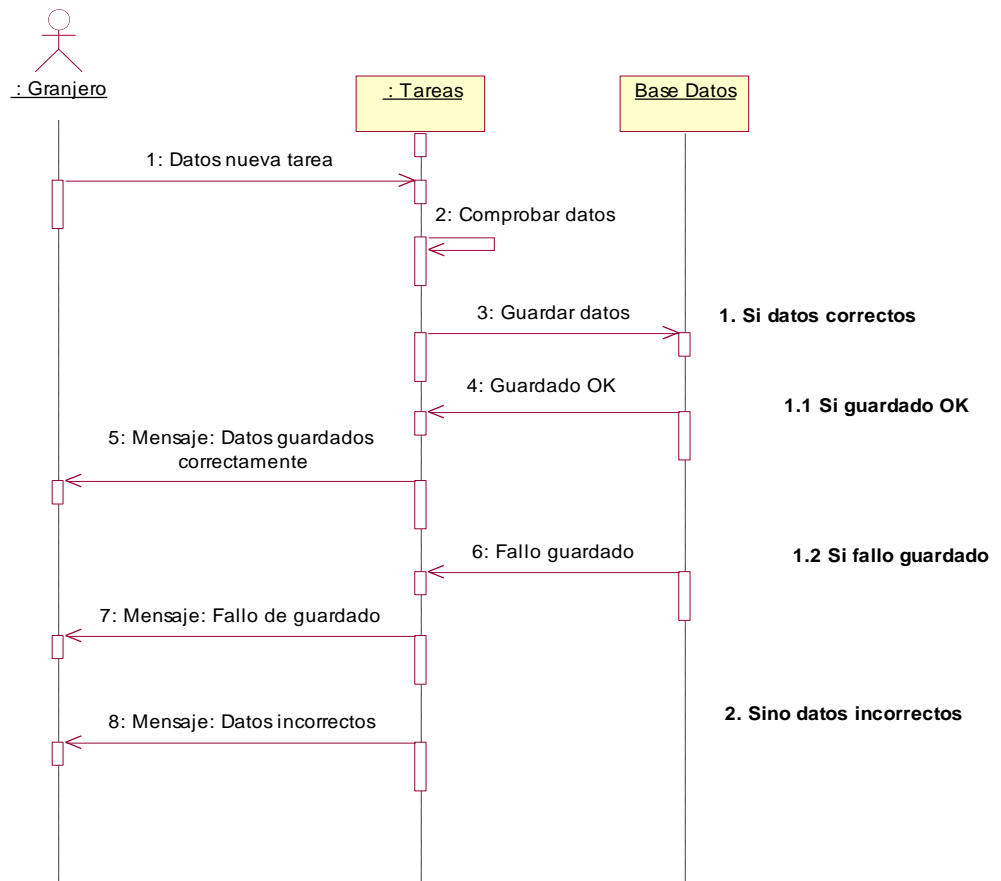


Imagen 28: Diagrama de secuencia del usuario Granjero para el caso de uso NuevaTarea.

9. Diagrama de estados:

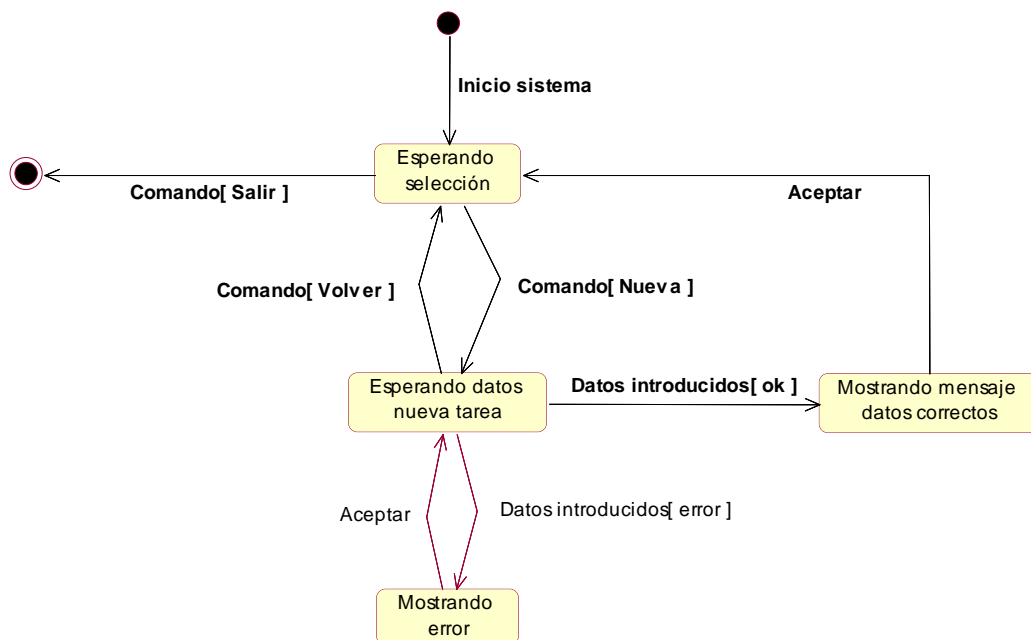


Imagen 29: Diagrama de estados para el caso de uso NuevaTarea.

- **Caso de uso: SeleccionarGranja**

1. Descripción: Debemos de consultar las granjas que se encuentran activas en el sistema y de las que, por tanto, se pueden modificar sus componentes.
2. Actores: Instalador, AdministradorTelecentro, Granjero
3. Precondiciones:
4. Pasos:
 1. Seleccionamos una de las granjas
 2. Confirmamos la selección.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

1. AdministradorTelecentro

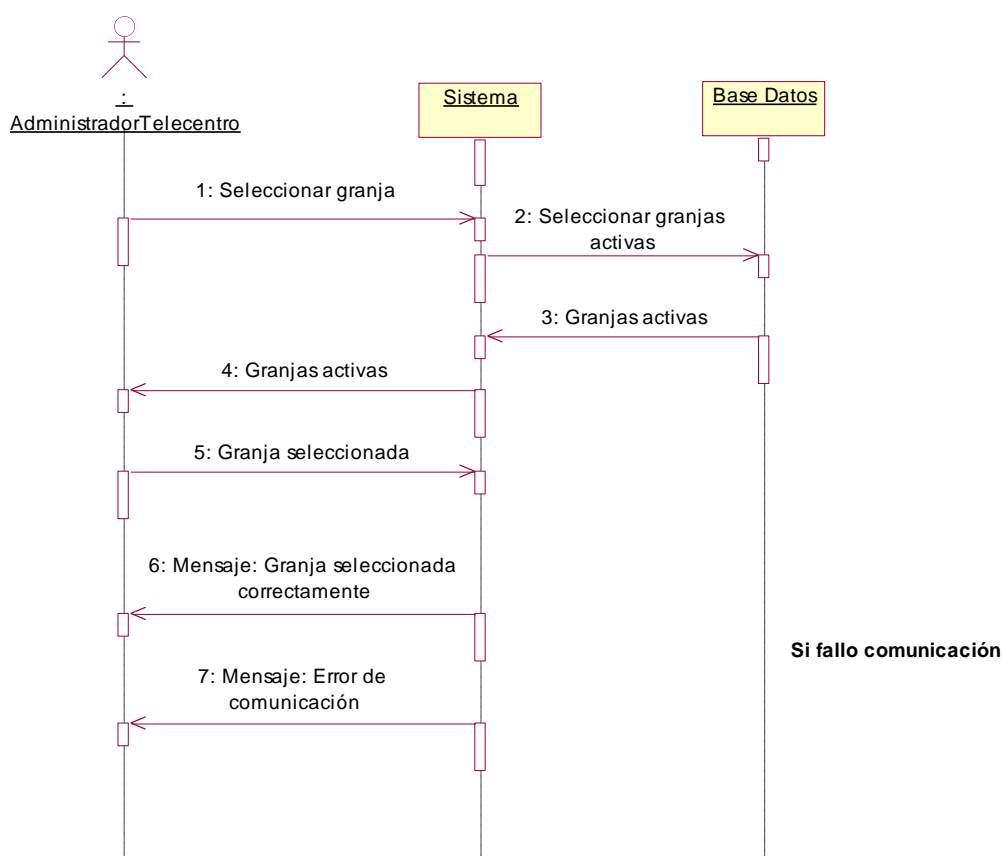


Imagen 30: Diagrama de secuencia del usuario AdministradorTelecentro para el caso de uso SeleccionarGranja.

Nota: La secuencia la realizo con Sistema ya que se realizan los mismos pasos para la clase VisualizaciónGranja como para la clase PantallaConfiguracion, de este modo no pongo gráficos iguales.

2. Instalador

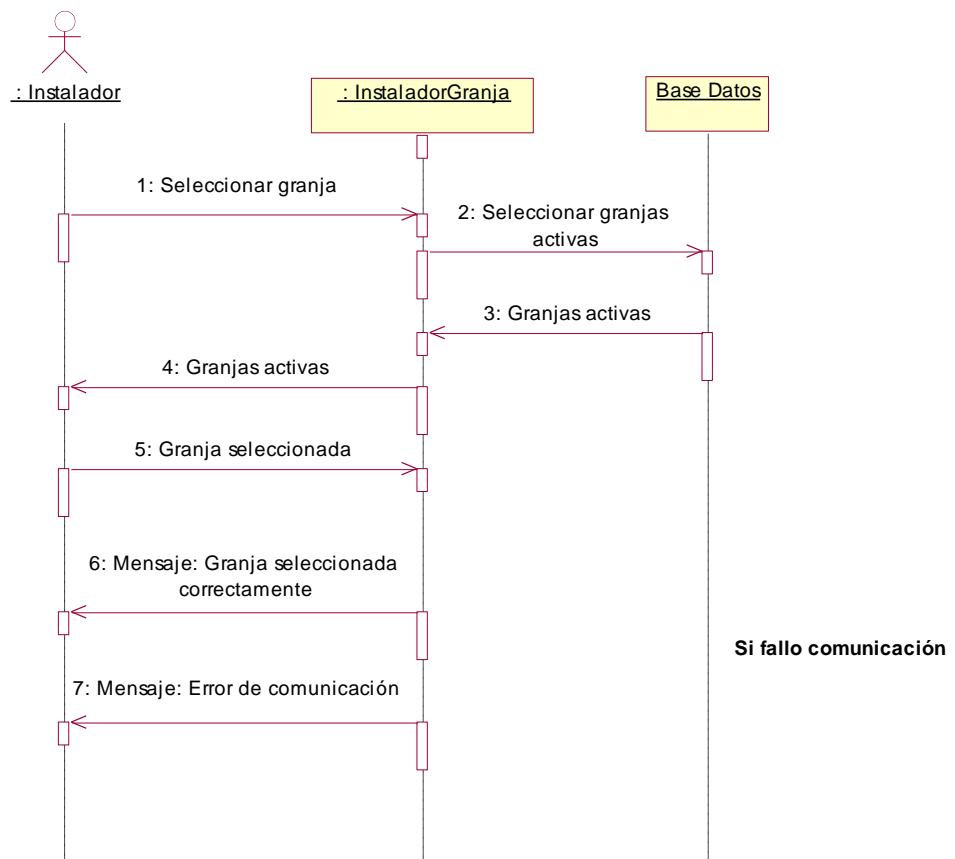


Imagen 31: Diagrama de secuencia del usuario Instalador para el caso de uso SeleccionarGranja.

3. Granjero

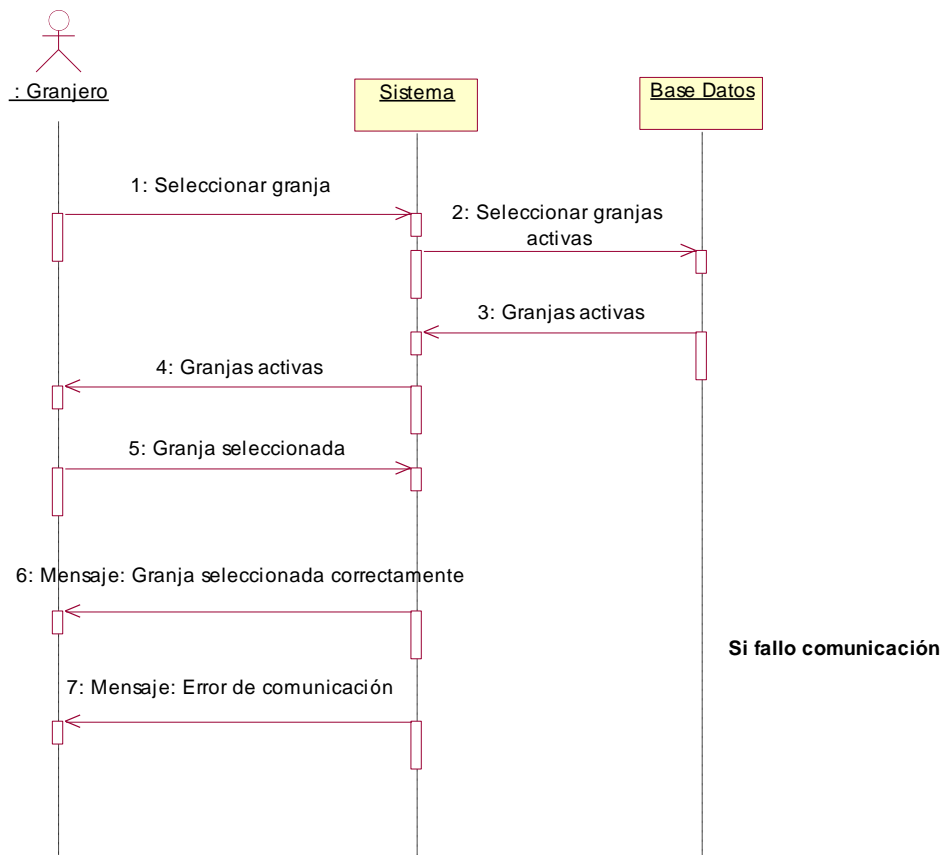


Imagen 32: Diagrama de secuencia del usuario Granjero para el caso de uso SeleccionarGranja.

Nota: La secuencia la realizo con Sistema ya que se realizan los mismos pasos para la clase VisualizaciónGranja como para la clase PantallaConfiguracion, de este modo no pongo gráficos iguales.

9. Diagrama de estados:

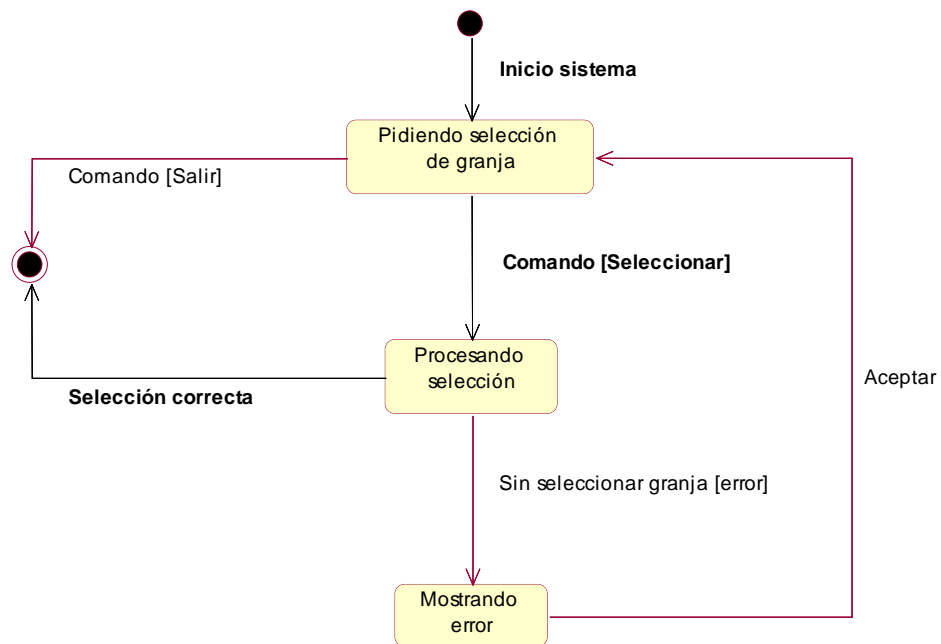


Imagen 33: Diagrama de estados para el caso de uso SeleccionarGranja.

Diagrama de casos de uso para el usuario Instalador

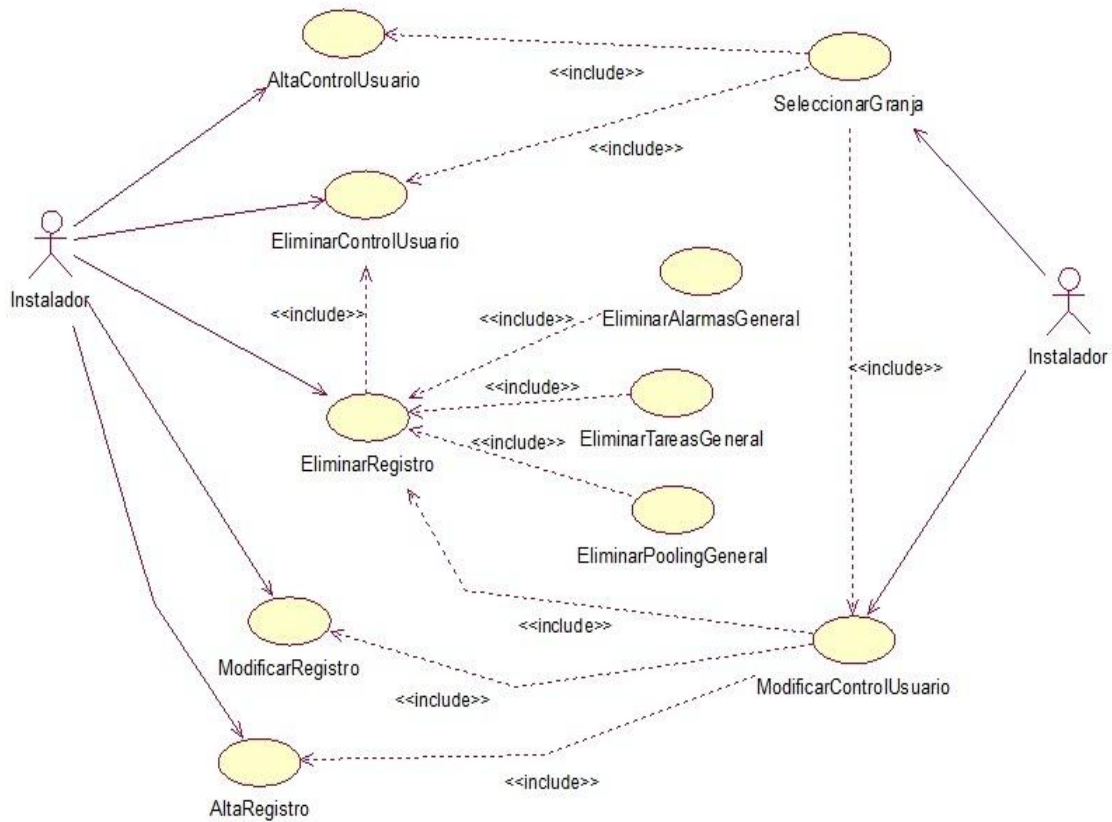


Imagen 34: Diagrama de casos de uso para el usuario Instalador

En la Imagen 34 podemos observar los diferentes casos de uso que se han analizado en el sistema para el usuario Instalador. Estas son todas las acciones que puede llevar a cabo este usuario según el análisis que se ha realizado del sistema. A continuación se explica cada uno de los casos de uso por separado, mostrando además los diagramas de secuencia y de estados que tienen asociados.

- **Caso de uso: AltaControlUsuario**

1. Descripción: Alta de un control de usuario, que va a ser el objeto del sistema que se encargue de manejar los elementos físicos conectados en la instalación agropecuaria.
2. Actores: Instalador
3. Precondiciones:
 1. Se debe de haber seleccionado una granja de la consulta de granjas activas.
4. Pasos:
 1. Introducimos los datos del nuevo control de usuario.
 2. El sistema valida que los datos sean correctos.
 3. Si los datos son correctos mostramos un mensaje de que los datos se han guardado correctamente.
 4. Si los datos no son correctos mostramos un aviso para que modifique los datos incorrectos.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

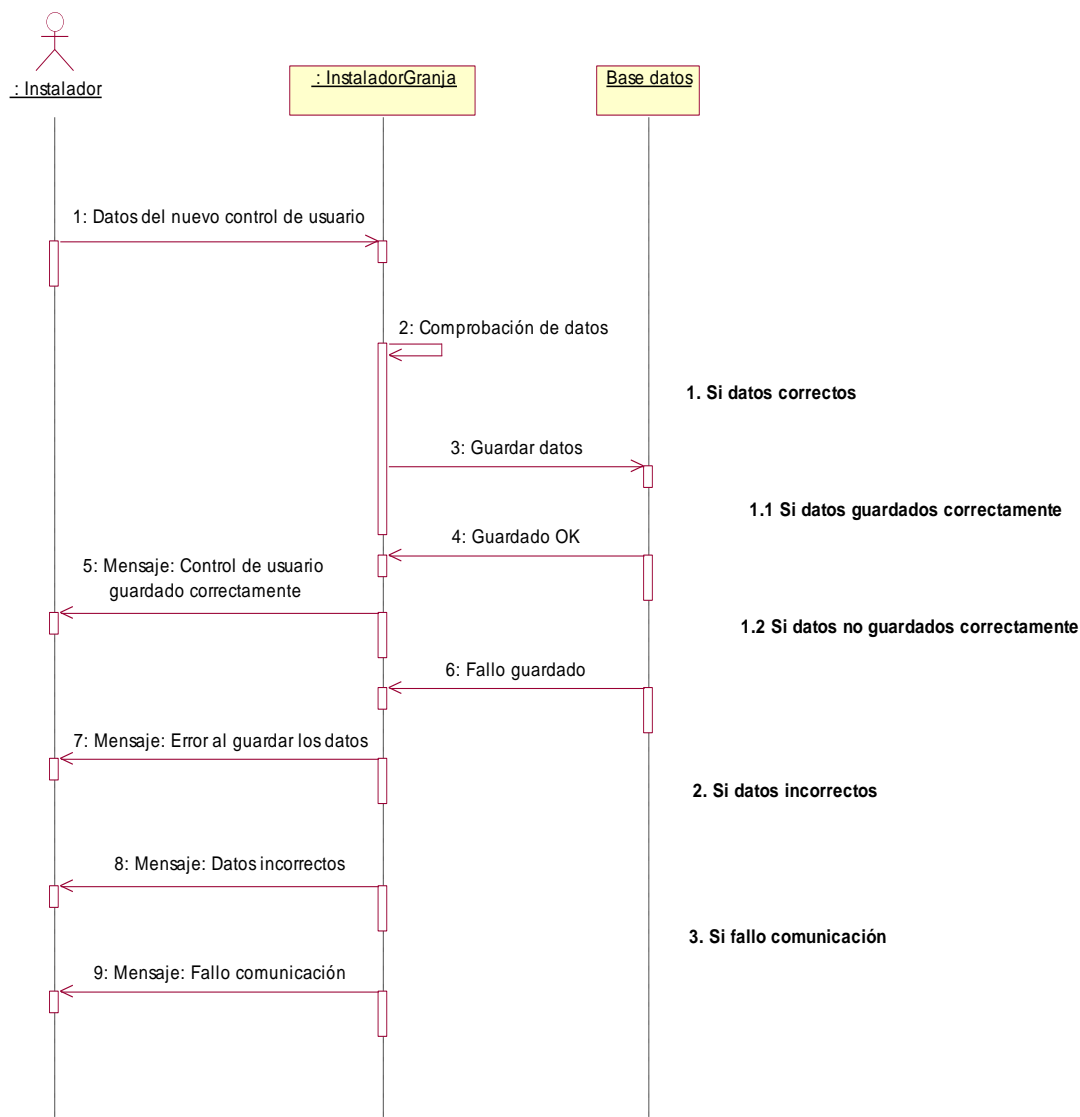


Imagen 35: Diagrama de secuencia del usuario Instalador para el caso de uso AltaControlUsuario.

9. Diagrama de estados:

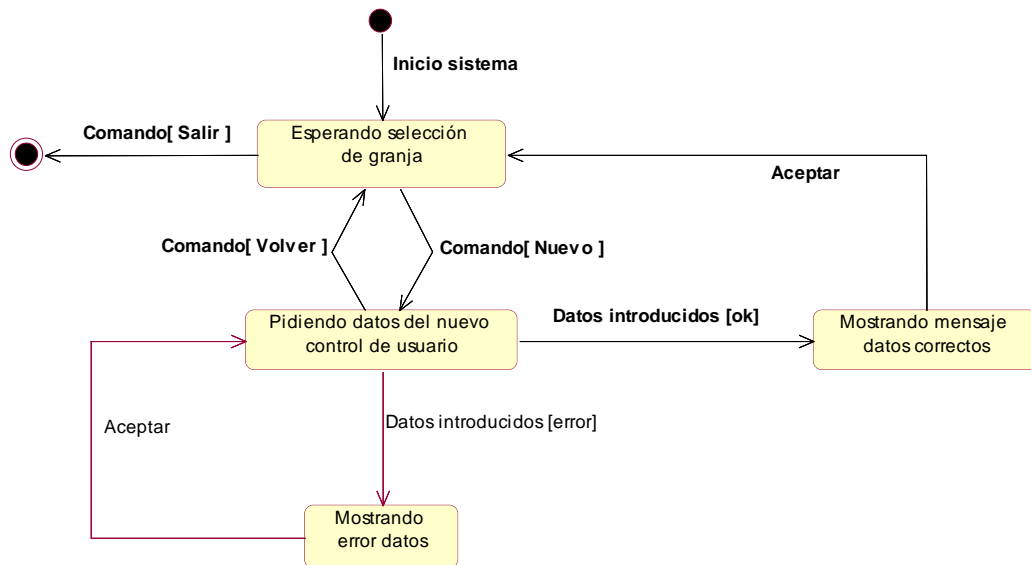


Imagen 36: Diagrama de estados para el caso de uso AltaControlUsuario

- **Caso de uso: EliminarControlUsuario**

1. Descripción: Eliminación de un control de usuario asociado a una granja.
2. Actores: Instalador
3. Precondiciones:
 1. Se debe de haber seleccionado una granja de la consulta de granjas activas.
 2. Se debe de haber seleccionado el control de usuario que se desea eliminar.
4. Pasos:
 1. Al eliminar el control de usuario se eliminan todos los registros que tiene asociados ese control de usuario así como todas las programaciones que tienen asociadas los registros.
 2. Si la eliminación se ha realizado correctamente se muestra un mensaje.
 3. Si ha ocurrido algún error en la eliminación se muestra un mensaje.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

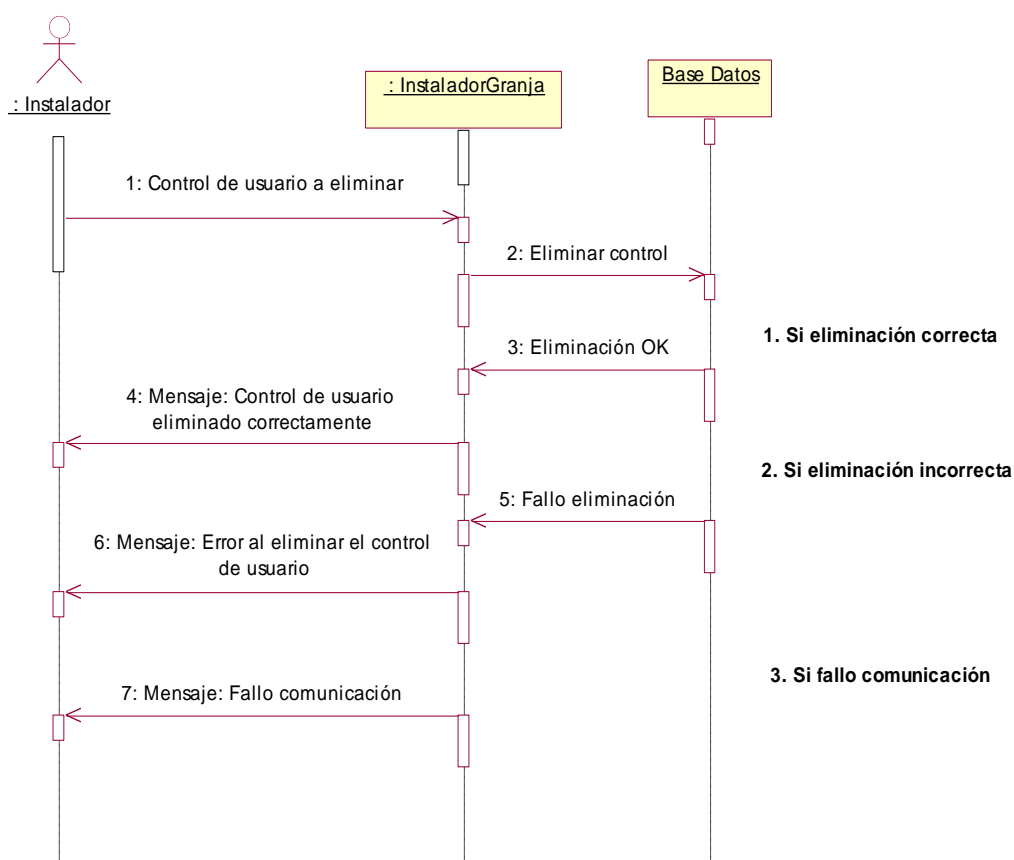


Imagen 37: Diagrama de secuencia del usuario Instalador para el caso de uso EliminarControlUsuario.

9. Diagrama de estados:

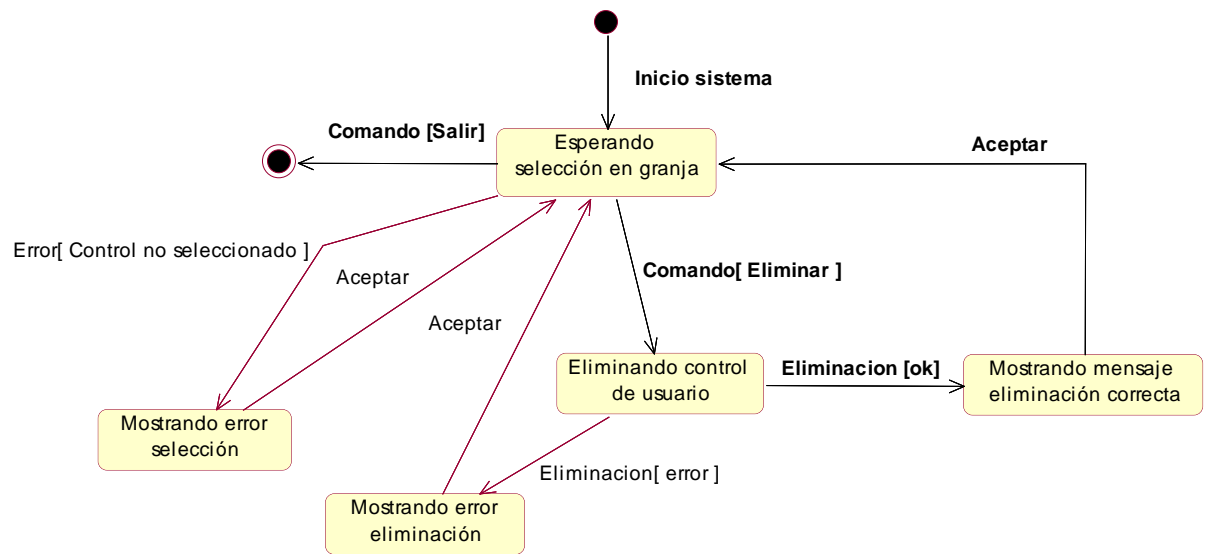


Imagen 38: Diagrama de estados para el caso de uso EliminarControlUsuario.

- **Caso de uso: EliminarRegistro**

1. Descripción: Eliminación de un registro que se encuentra asociado a un control de usuario de una granja.
2. Actores: Instalador
3. Precondiciones:
 1. Se debe de seleccionar la granja en la que se encuentra el registro.
 2. Se debe seleccionar el control de usuario que tiene asociado el registro que queremos eliminar.
 3. Se debe de seleccionar el registro que se quiere eliminar de entre la lista de registros asociados al control de usuario.
4. Pasos:
 1. Al eliminar un registro se eliminan además, según el tipo de registro que sea, las alarmas, las tareas y la configuración para el refresco que tenga asociados.
 2. Si la eliminación se ha realizado correctamente se muestra un mensaje.
 3. Si ha ocurrido algún error en la eliminación se muestra un mensaje.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

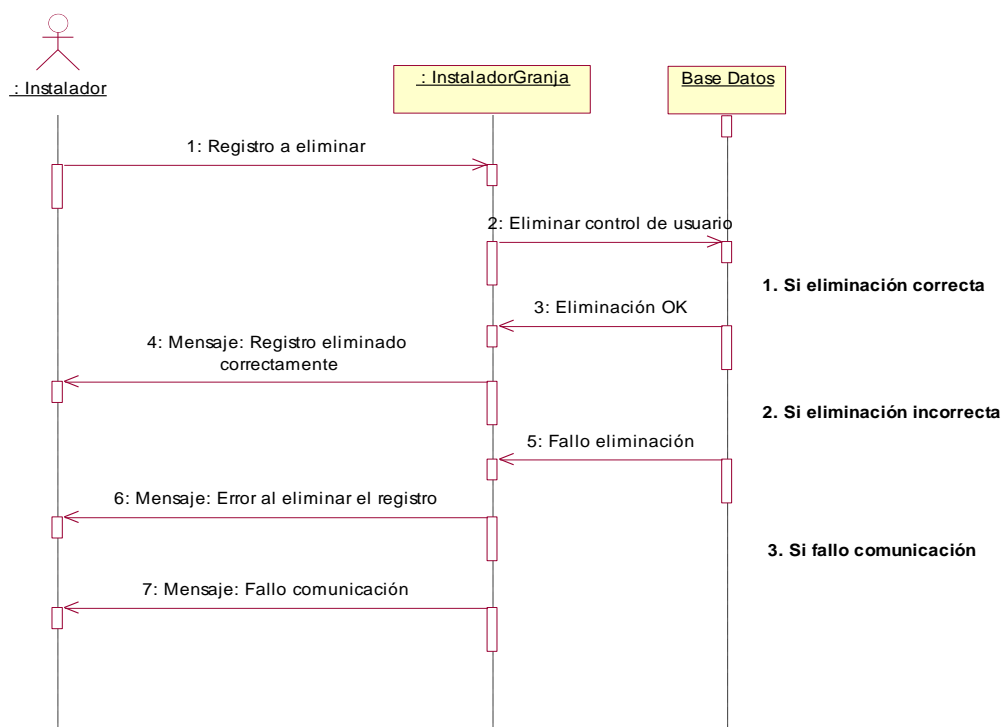


Imagen 39: Diagrama de secuencia del usuario Instalador para el caso de uso EliminarRegistro.

9. Diagrama de estados:

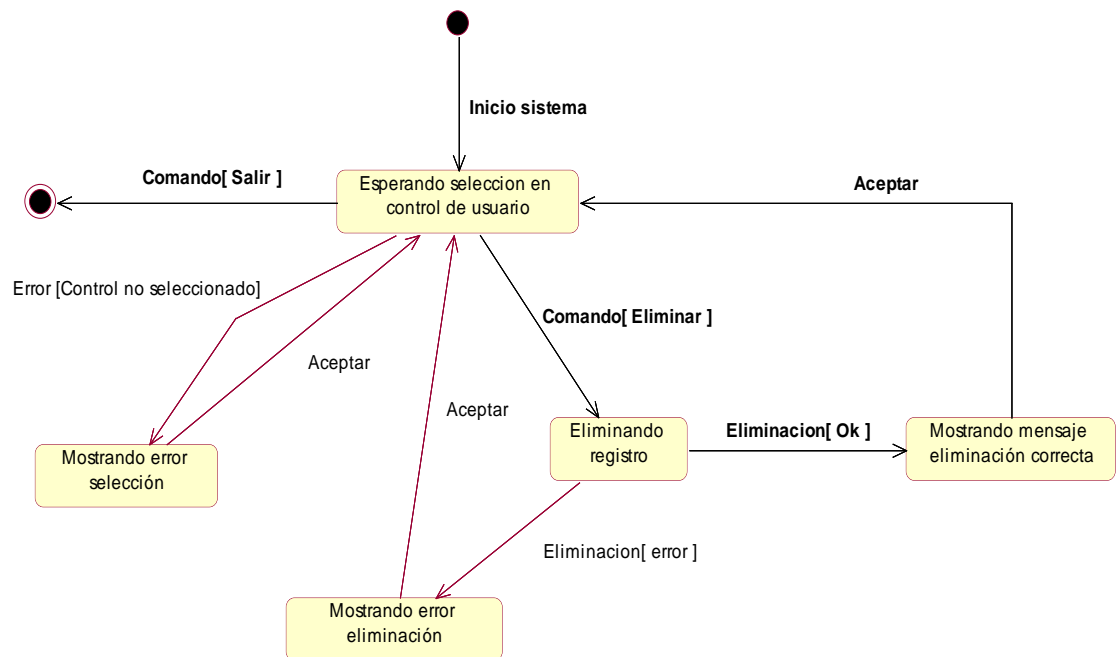


Imagen 40: Diagrama de estados para el caso de uso EliminarRegistro.

- **Caso de uso: ModificarRegistro**

1. Descripción: Modificar las propiedades características de un registro que compone un control de usuario.
2. Actores: Instalador
3. Precondiciones:
 1. Se debe de seleccionar la granja en la que se encuentra el registro.
 2. Se debe seleccionar el control de usuario que tiene asociado el registro que queremos eliminar.
 3. Se debe de seleccionar el registro que se quiere modificar de entre la lista de registros asociados al control de usuario.
4. Pasos:
 1. Introducimos los nuevos valores de las propiedades que se quieren modificar.
 2. Si la modificación se ha realizado correctamente se muestra un mensaje.
 3. Si ha ocurrido algún error en la modificación se muestra un mensaje.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

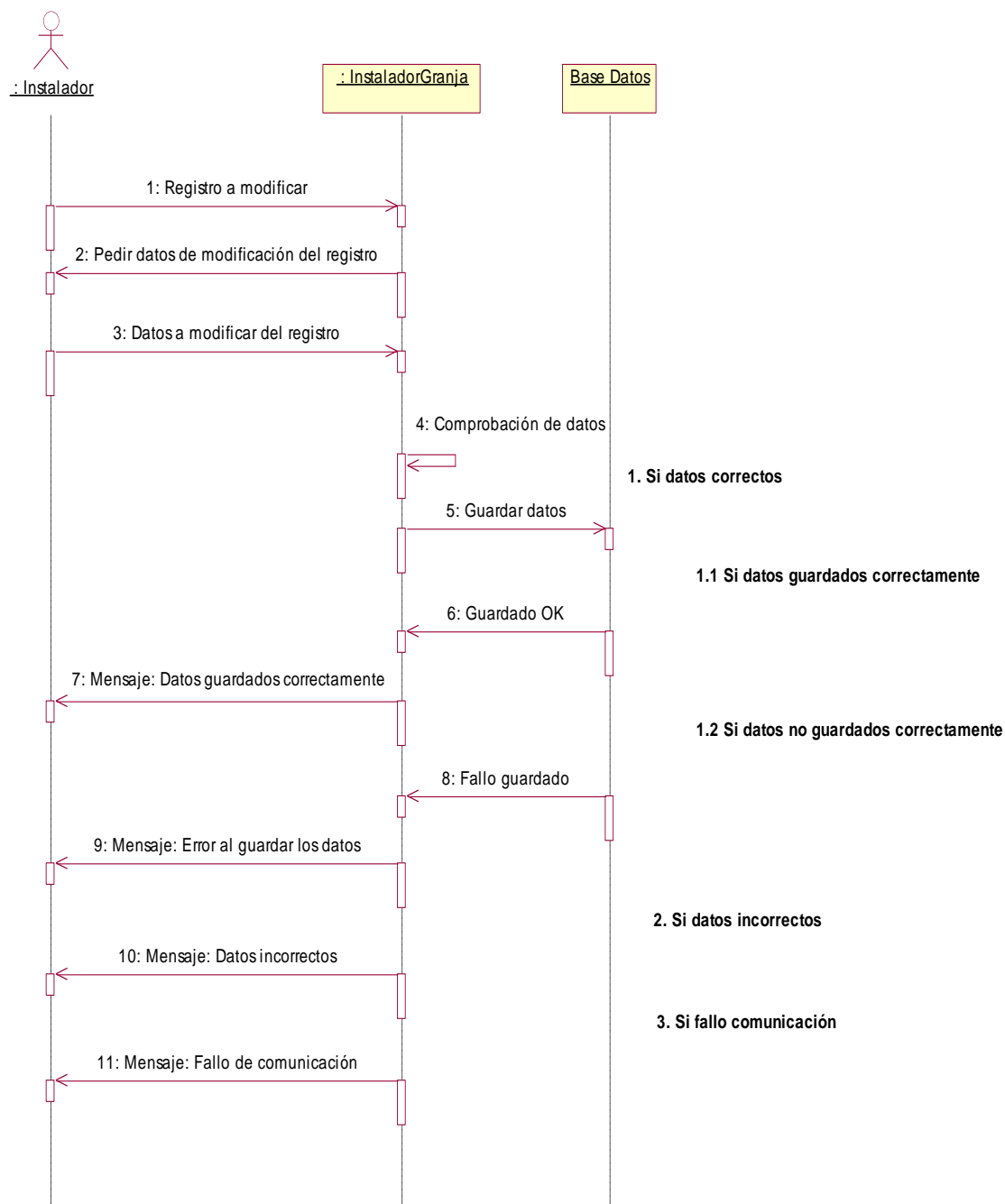


Imagen 41: Diagrama de secuencia del usuario Instalador para el caso de uso ModificarRegistro.

9. Diagrama de estados:

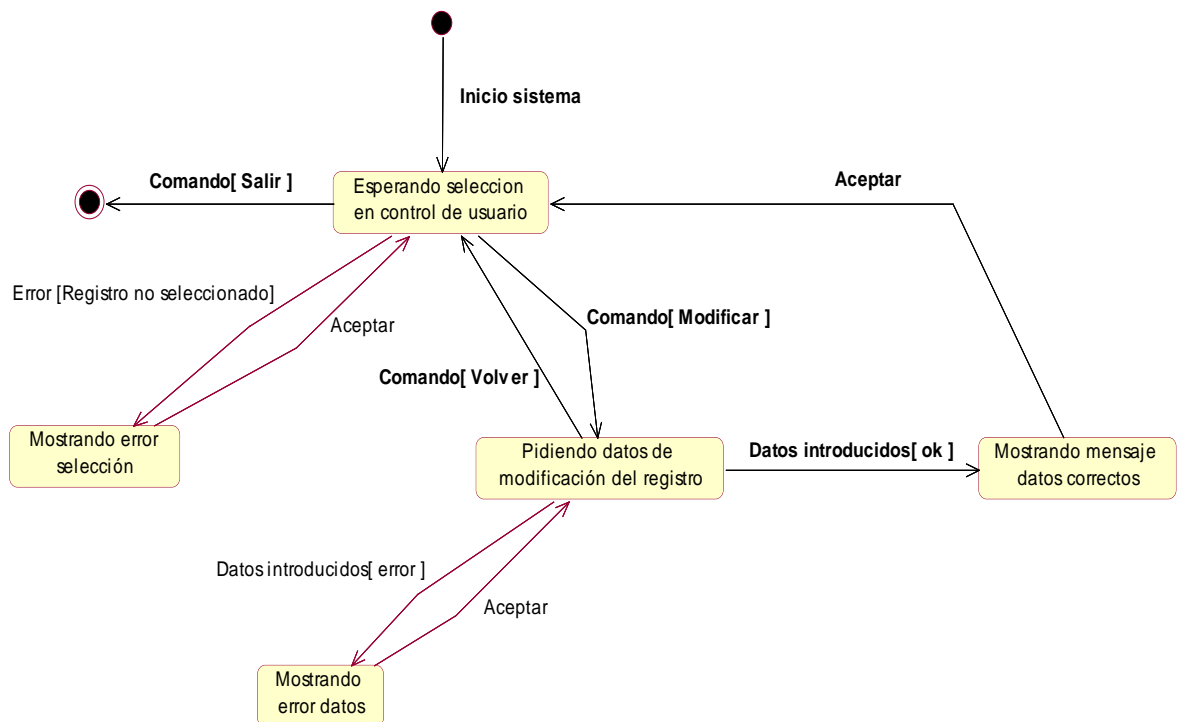


Imagen 42: Diagrama de estados para el caso de uso ModificarRegistro.

- **Caso de uso: AltaRegistro**

1. Descripción: Dar de alta un nuevo registro asociado a un control de usuario de una granja específica.
2. Actores: Instalador
3. Precondiciones:
 1. Se debe de seleccionar la granja en la que queremos dar de alta el nuevo registro.
 2. Se debe seleccionar el control de usuario que va a tener asociado al nuevo registro.
4. Pasos:
 1. Introducir los valores de las propiedades del nuevo registro.
 2. Si la modificación se ha realizado correctamente se muestra un mensaje.
 3. Si ha ocurrido algún error en la modificación se muestra un mensaje.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

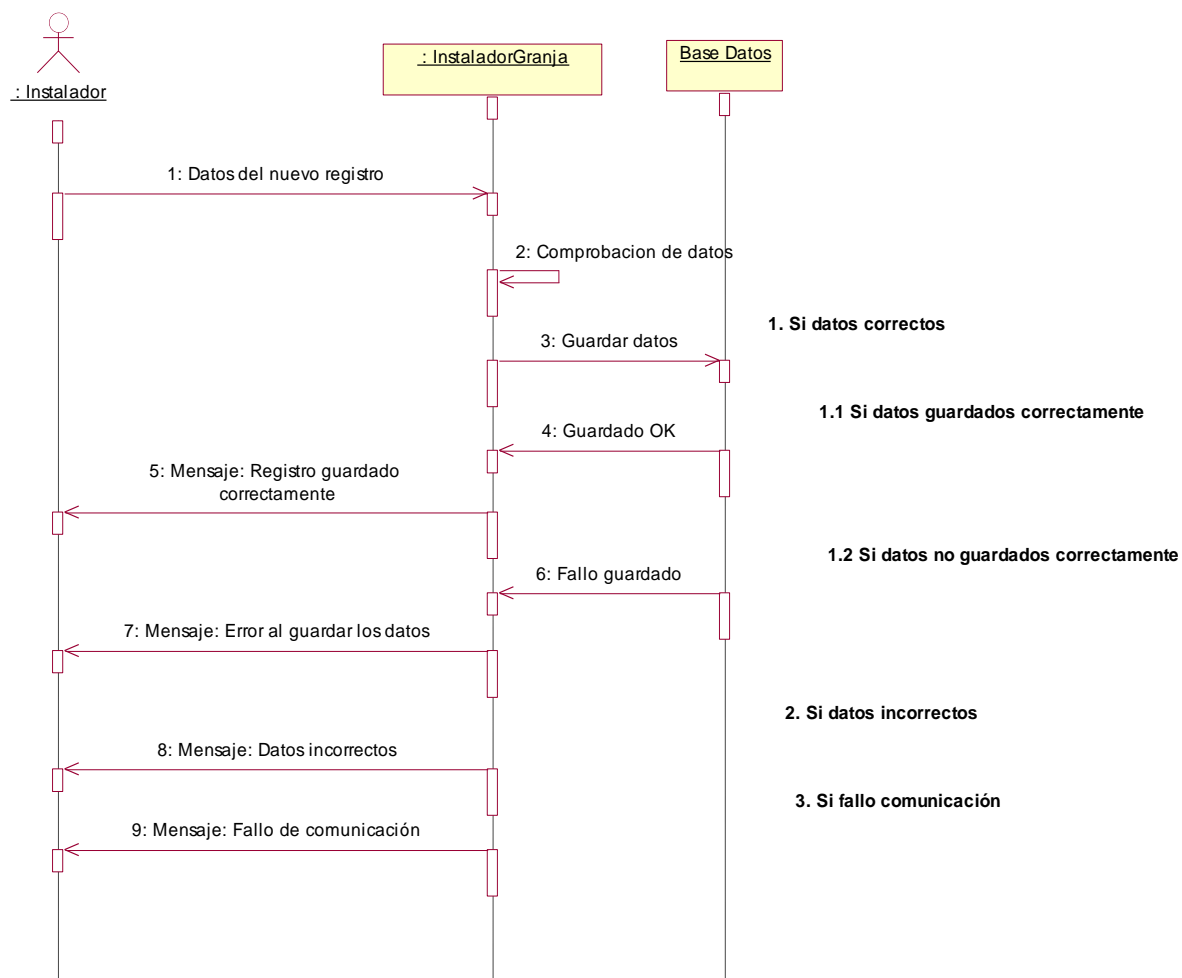


Imagen 43: Diagrama de secuencia del usuario Instalador para el caso de uso AltaRegistro.

9. Diagrama de estados:

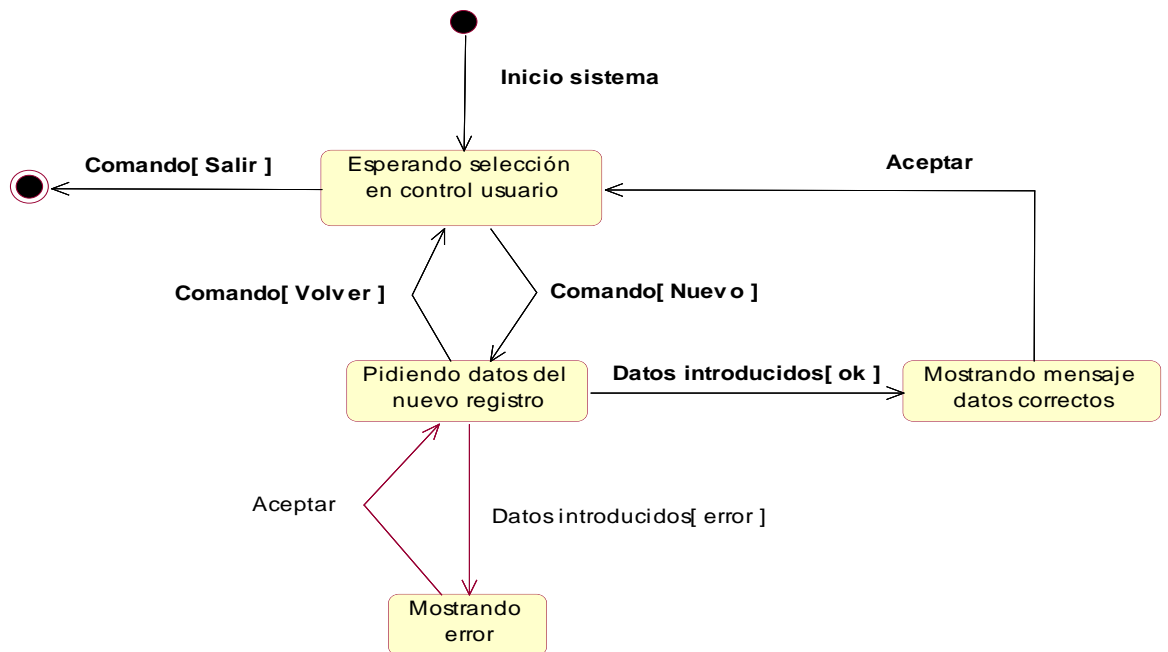


Imagen 44: Diagrama de estados para el caso de uso AltaRegistro.

- **Caso de uso: ModificarControlUsuario**

1. Descripción: Modificación de un control de usuario, para cambiar el valor de algunas de sus propiedades características.
2. Actores: Instalador
3. Precondiciones:
 1. Se debe de haber seleccionado una granja de la consulta de granjas activas.
 2. Se debe de haber seleccionado el control de usuario que se desea modificar.
4. Pasos:
 1. Modificamos los parámetros que sean necesarios.
 2. El sistema valida que los datos sean correctos.
 3. Si los datos son correctos mostramos un mensaje de que los datos se han guardado correctamente.
 4. Si los datos no son correctos mostramos un aviso para que modifique los datos incorrectos.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

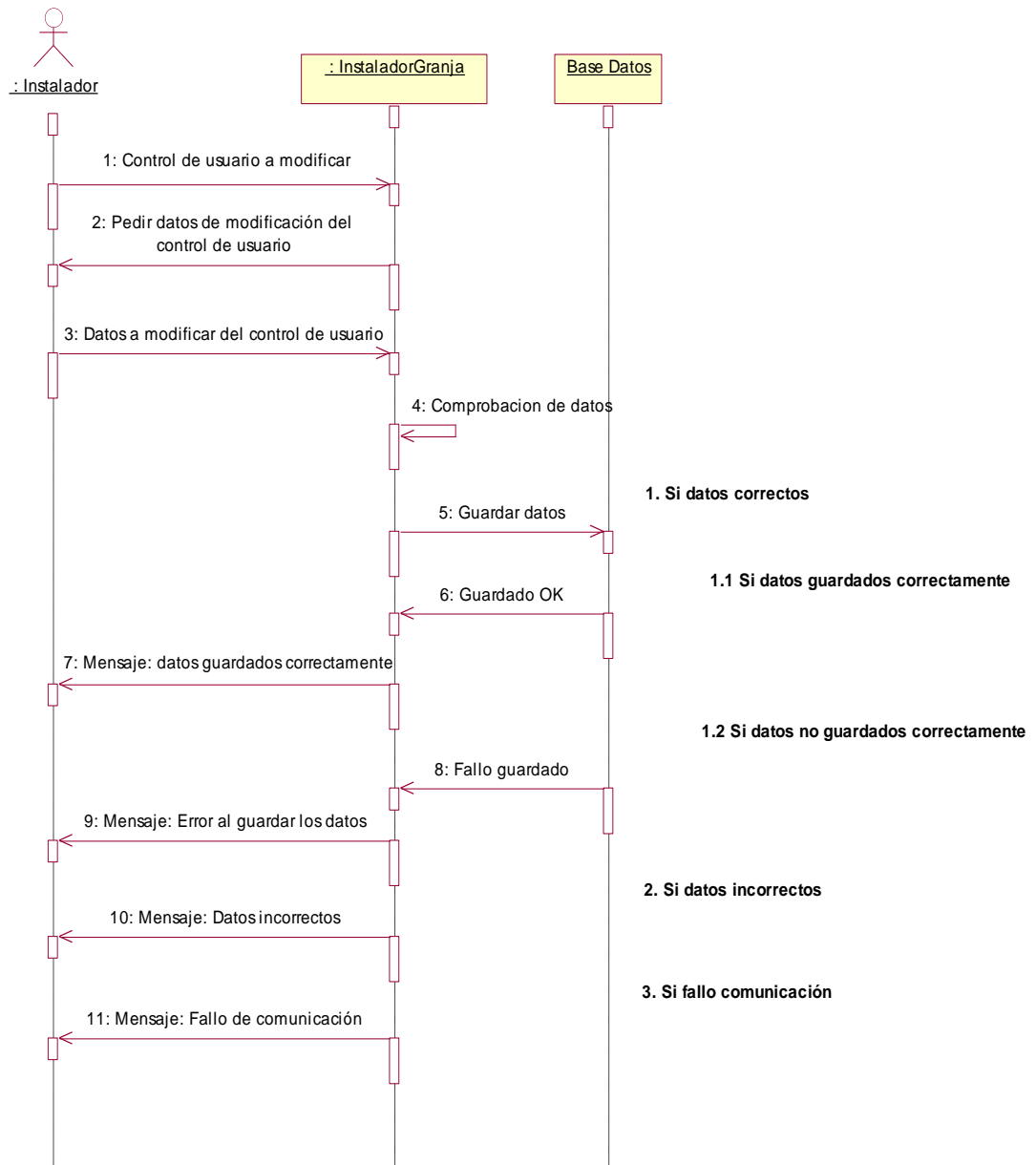


Imagen 45: Diagrama de secuencia del usuario Instalador para el caso de uso ModificarControlUsuario.

9. Diagrama de estados:

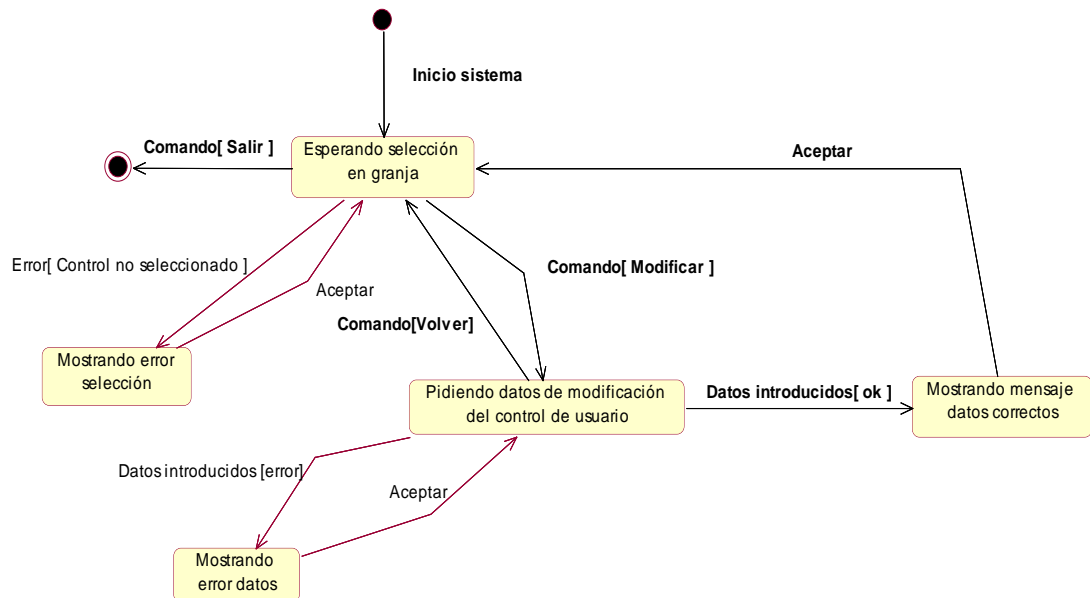


Imagen 46: Diagrama de estados para el caso de uso ModificarControlUsuario.

- **Caso de uso: EliminarAlarmasGeneral**

1. Descripción: Eliminación de la alarma que tenga programada un registro. El registro debe de ser un registro de salida para el sistema, es decir, el registro nos envía información al sistema sobre su funcionamiento.
2. Actores: Instalador, AdministradorTelecentro
3. Precondiciones:
4. Pasos:
 1. Eliminar todas las alarmas que pertenezcan al registro especificado.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

1. Instalador

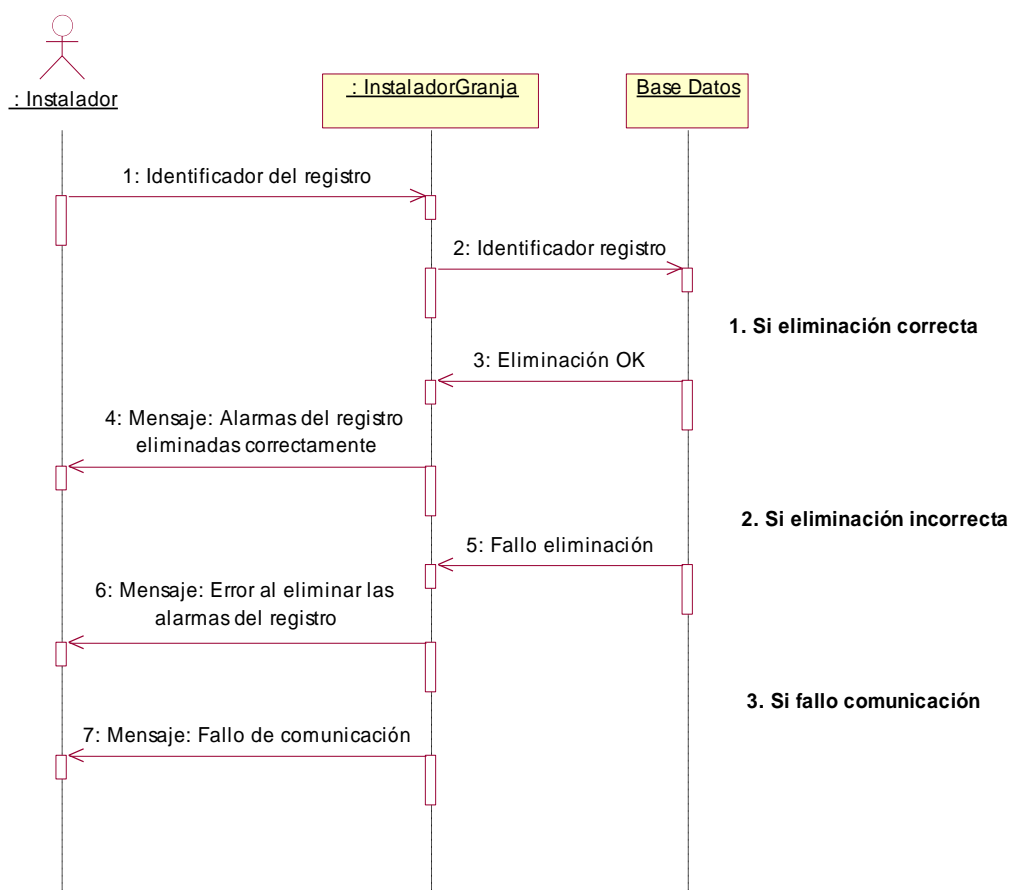


Imagen 47: Diagrama de secuencia del usuario Instalador para el caso de uso EliminarAlarmasGeneral.

2. AdministradorTelecentro

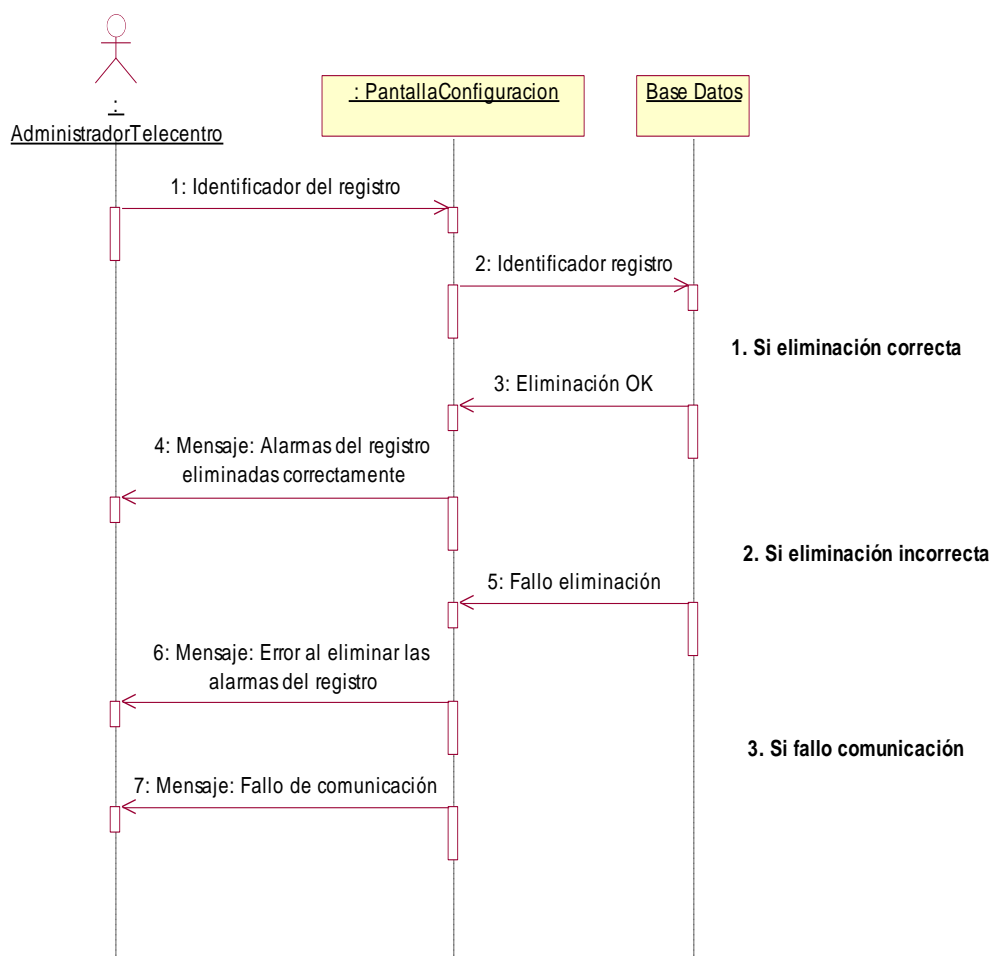


Imagen 48: Diagrama de secuencia del usuario AdministradorTelecentro para el caso de uso EliminarAlarmasGeneral.

9. Diagrama de estados:

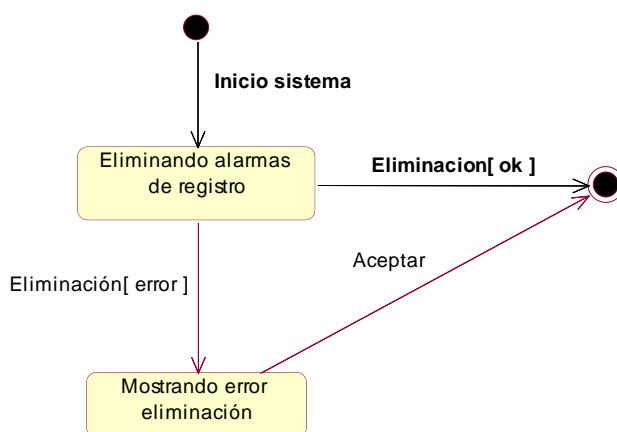


Imagen 49: Diagrama de estados para el caso de uso EliminarAlarmasGeneral.

- **Caso de uso: EliminarTareasGeneral**

1. Descripción: Eliminación de la tarea programada que tenga un registro. El registro debe de ser un registro de entrada para el sistema, es decir, al registro enviamos información para su funcionamiento.
2. Actores: Instalador, AdministradorTelecentro
3. Precondiciones:
4. Pasos:
 1. Eliminar todas las tareas que pertenezcan al registro especificado.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

1. Instalador

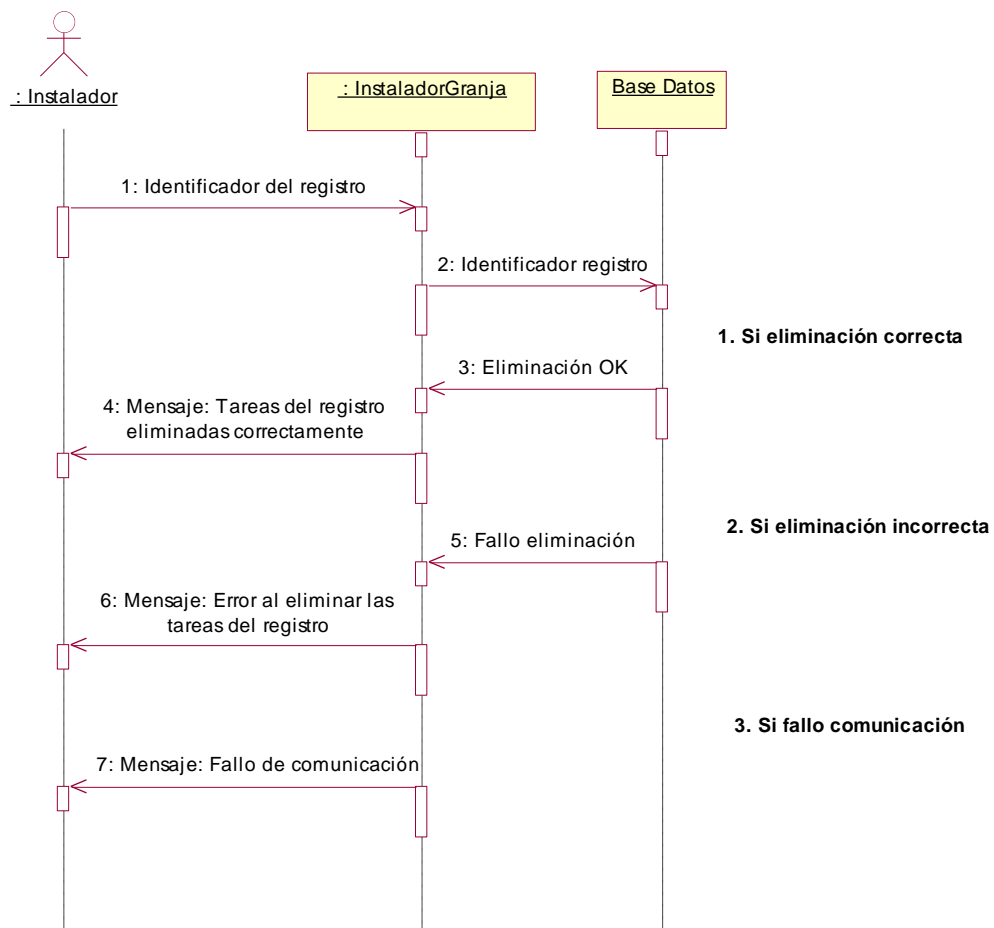


Imagen 50: Diagrama de secuencia del usuario Instalador para el caso de uso EliminarTareasGeneral.

2. AdministradorTelecentro

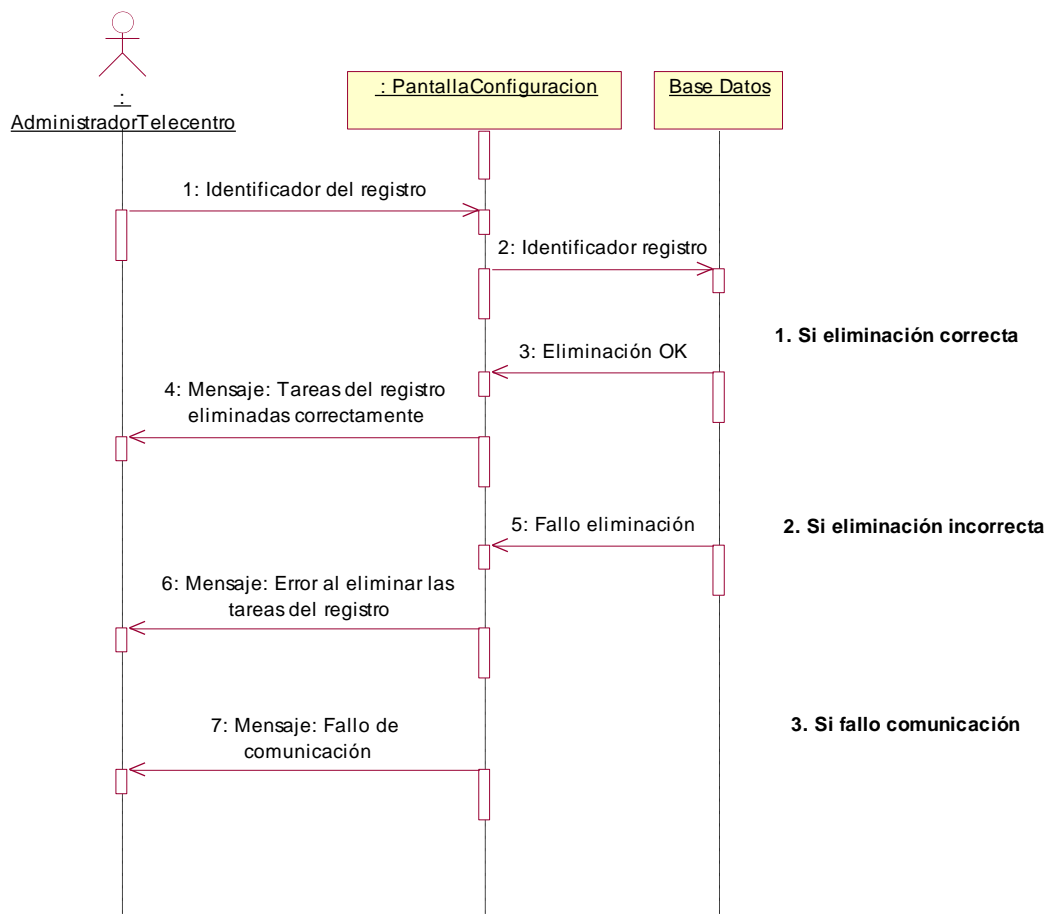


Imagen 51: Diagrama de secuencia del usuario AdministradorTelecentro para el caso de uso EliminarTareasGeneral.

9. Diagrama de estados:

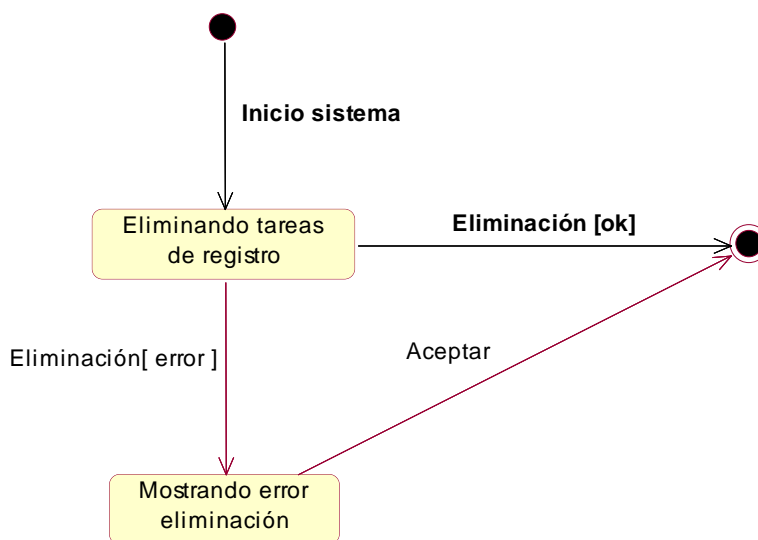


Imagen 52: Diagrama de estados para el caso de uso EliminarTareasGeneral.

- **Caso de uso: EliminarPoolingGeneral**

1. Descripción: Eliminación de las configuraciones de refresco que tenga programadas un registro. El registro debe de ser un registro de salida para el sistema, es decir, el registro nos envía información al sistema sobre su funcionamiento
2. Actores: Instalador
3. Precondiciones:
4. Pasos:
 1. Eliminar el tiempo de pooling del registro seleccionado.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
8. Diagrama de secuencia:

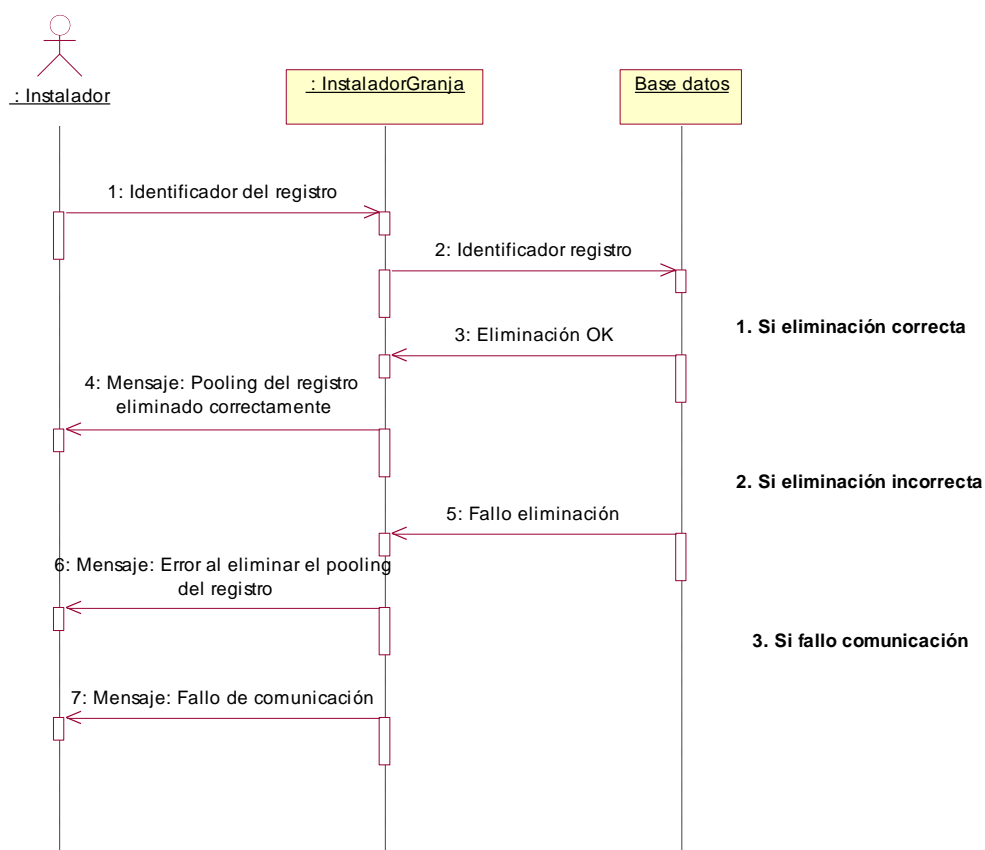


Imagen 53: Diagrama de secuencia del usuario Instalador para el caso de uso EliminarPoolingGeneral.

9. Diagrama de estados:

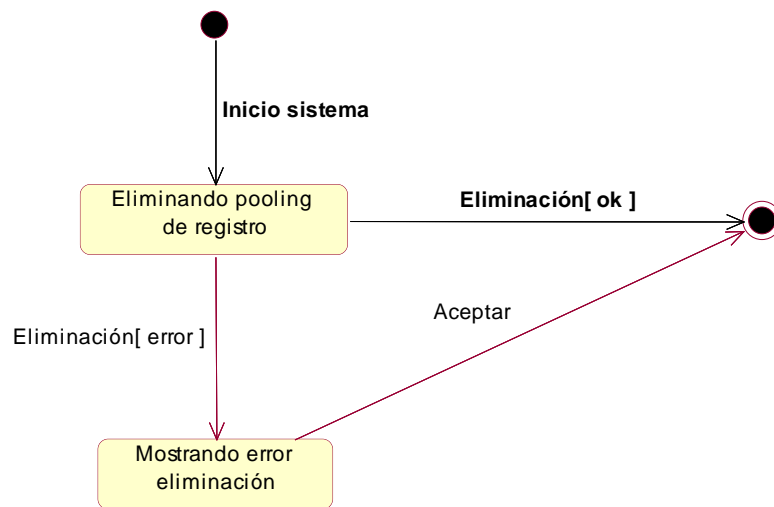


Imagen 54: Diagrama de estados para el caso de uso EliminarPoolingGeneral.

Diagrama de casos de uso para el usuario AdministradorTelecentro

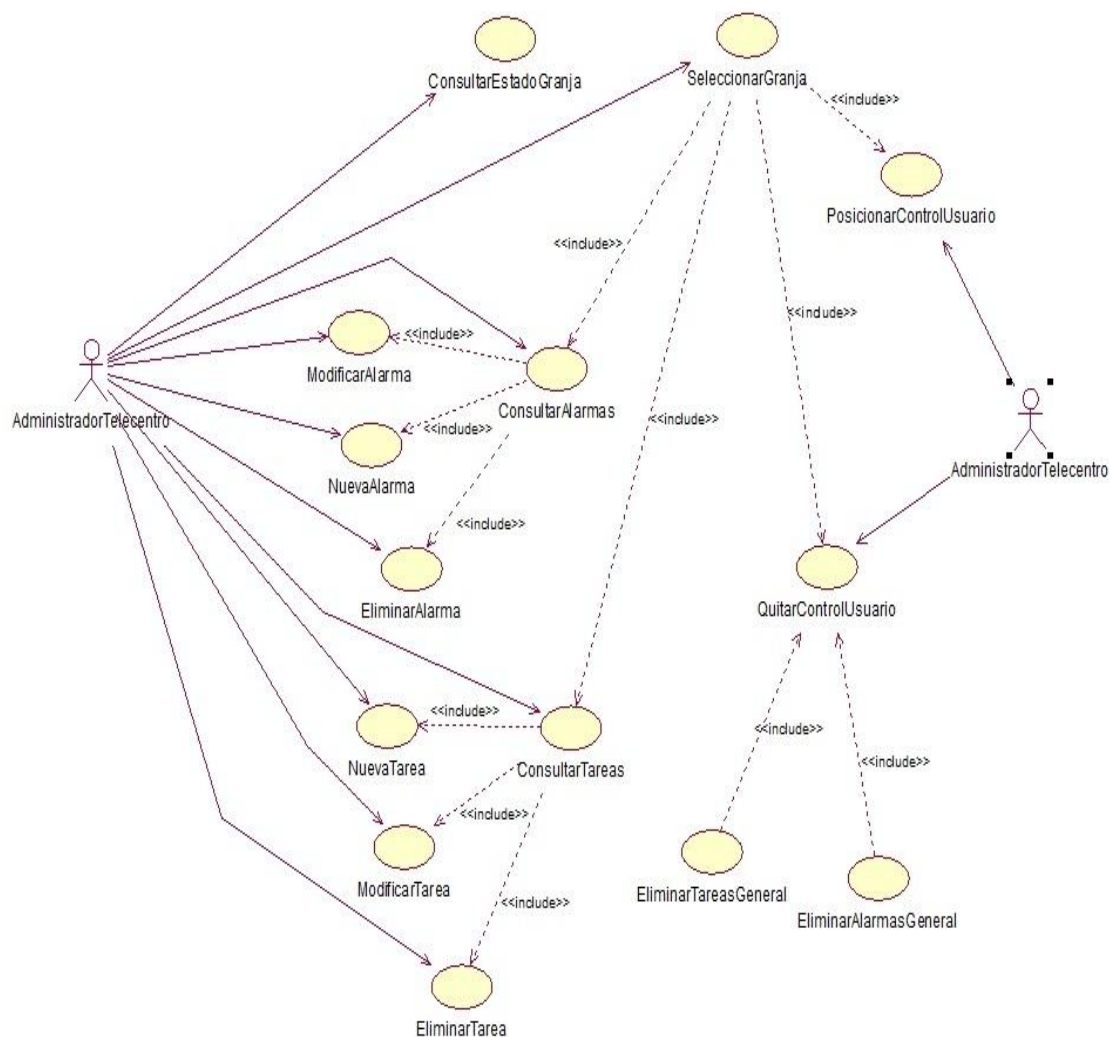


Imagen 55: Diagrama de casos de uso para el usuario AdministradorTelecentro

En la Imagen 55 podemos observar los diferentes casos de uso que se han analizado en el sistema para el usuario Instalador. Estas son todas las acciones que puede llevar a cabo este usuario según el análisis que se ha realizado del sistema. A continuación se explica cada uno de los casos de uso por separado, mostrando además los diagramas de secuencia y de estados que tienen asociados (la mayoría de los casos de uso ya han sido especificados anteriormente por lo que no los voy a repetir).

- **Caso de uso: PosicionarControlUsuario**

1. Descripción: Para una granja en concreto, debemos de ser capaces de seleccionar los controles de usuario que tiene instalados y posicionarlos, cada uno por separado, en un punto concreto de la pantalla. Cada control de usuario representará un elemento físico que se encuentra instalado en la granja (interruptor de bombilla, termómetro...).
2. Actores: AdministradorTelecentro
3. Precondiciones:
 1. Se debe seleccionar previamente una granja dada de alta en el sistema.
4. Pasos:
 1. Seleccionar el control de usuario de entre los controles de usuario que pertenecen a la granja.
 2. Arrastrarlo hasta el punto de la pantalla en el que se quiere posicionar.
 3. Al quedar posicionado en la pantalla deberá de eliminarse de la barra lateral en la que estaba ubicado antes de ser arrastrado.
 4. Guardar los cambios para mantener la posición de control.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
 1. El control de usuario va a tener una representación gráfica que será la que se mostrará al ser posicionado en la pantalla. Mientras no se encuentre posicionado deberá de permanecer ubicado en una barra lateral, clasificado según el tipo de control que sea y tendrá un icono y una descripción que le representará.
8. Diagrama de secuencia:

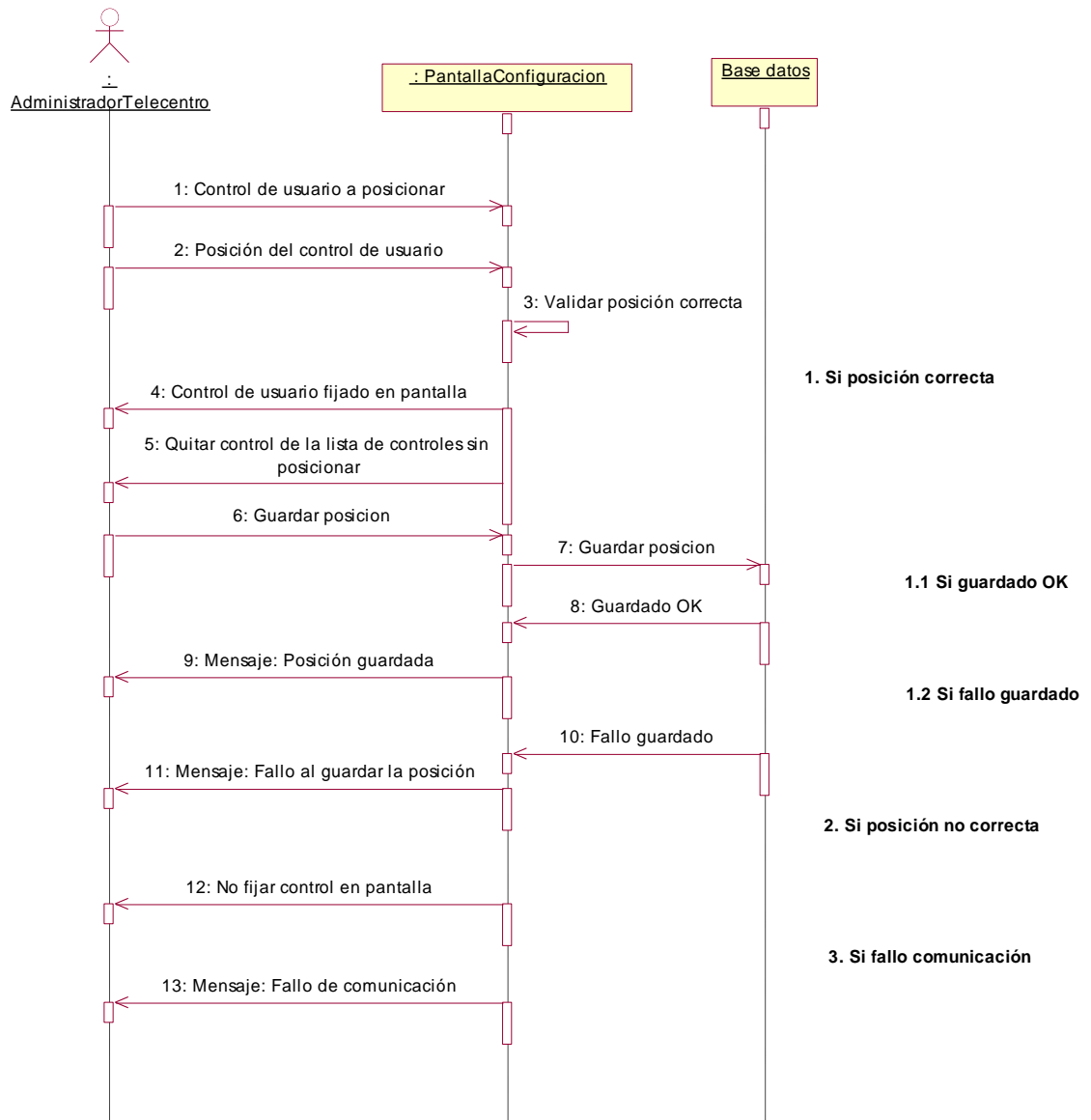


Imagen 56: Diagrama de secuencia del usuario AdministradorTelecentro para el caso de uso PosicionarControlUsuario.

- **Caso de uso: QuitarControlUsuario**

1. Descripción: Para una granja en concreto, debemos de ser capaces de quitar los controles de usuario que se encuentren posicionados en la pantalla. Para ello se deberán arrastrar desde su posición en la pantalla hasta un punto que se simbolizará como una papelera. Al ser eliminados de la pantalla se deberán posicionar en la barra lateral clasificados según el tipo de control.
2. Actores: AdministradorTelecentro
3. Precondiciones:
 1. Se debe seleccionar previamente una granja dada de alta en el sistema.
4. Pasos:
 1. Seleccionar el control de usuario que se quiere quitar.
 2. Arrastrarlo hasta la zona de la papelera.
 3. Guardar los cambios.
5. Variaciones:
6. Requisitos no funcionales:
7. Cuestiones:
 1. Cuando el control se deja en la papelera este se debe de agregar a la barra lateral donde se encuentran el resto de controles que aun no han sido arrastrados a la pantalla. La imagen del control deberá ser ahora el icono y la descripción que representan al control.
8. Diagrama de secuencia:

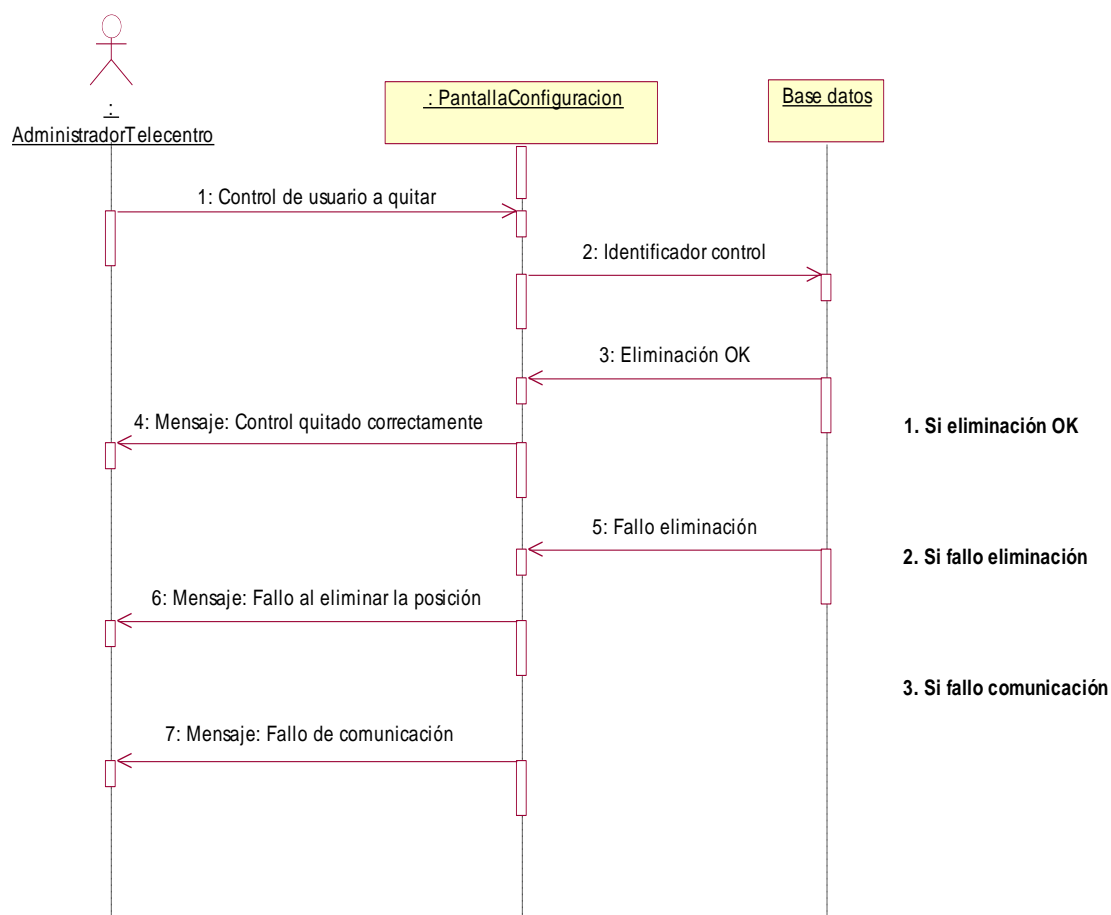


Imagen 57: Diagrama de secuencia del usuario AdministradorTelecentro para el caso de uso QuitarControlUsuario.

Capítulo 3

Diseño

En este capítulo voy a explicar cómo se ha realizado el diseño de la aplicación, a partir del análisis realizado previamente. En mi caso, la aplicación va a tener que acceder a datos contenidos en una base de datos, por lo que separaré esta funcionalidad en una capa independiente, para así independizar esta parte del resto de la aplicación. Además, es conveniente que se separen las interfaces de la aplicación de la funcionalidad de la misma, para así poder dividir el trabajo.

3.1 Diseño del sistema

3.1.1 Arquitectura del sistema software

En la elección de la arquitectura del sistema software se han sopesado las distintas opciones en favor de conseguir un sistema que fuese fácil de mantener, favoreciera la reutilización de código, poder separar la interfaz de la implementación, mejorar la cohesión de elementos semejantes y que fuese portable a todas las plataformas posibles. Todas estas características, en definitiva, son propiedades que deben de cumplirse para tener un software de calidad.

Por todo ello se ha optado por un patrón arquitectónico de tres capas:

- Capa de gestión de datos: se encuentra en el nivel más bajo de abstracción y es responsable de interactuar con el sistema gestor de bases de datos o con el sistema gestor de ficheros. En nuestro caso debe interactuar con el sistema gestor de bases de datos SQL Server.
- Capa de dominio: capa de nivel intermedio encargada de implementar las funcionalidades del sistema.
- Capa de presentación: es la capa con el nivel más alto de abstracción y cercana al usuario. Es responsable de la interacción con el usuario del sistema software.

A continuación veremos el diseño de cada una de las capas.

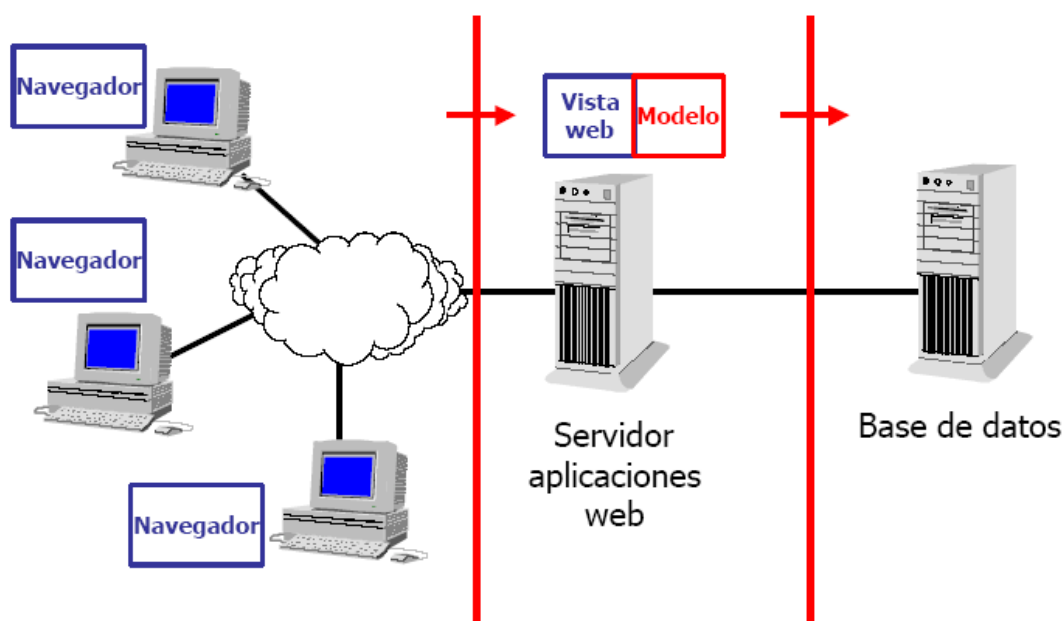


Imagen 58: Diseño de capas del proyecto

3.1.2 Diseño de la capa de gestión de datos

La capa de gestión de datos va a ser la encargada de realizar las comunicaciones entre la aplicación y la base de datos. En esta capa se deberán de crear todas las funciones necesarias para el acceso a datos. Pero lo primero que debemos de realizar es un buen análisis de las entidades que van a participar en el sistema y que van a necesitar tener almacenados datos en la base de datos. A partir de estas entidades vamos a construir en diagrama Entidad-Relación (Imagen 59), a partir del cual obtendremos unas tablas, que después de normalizarlas, se convertirán en las tablas de la base de datos.

1. Diagrama Entidad-Relación

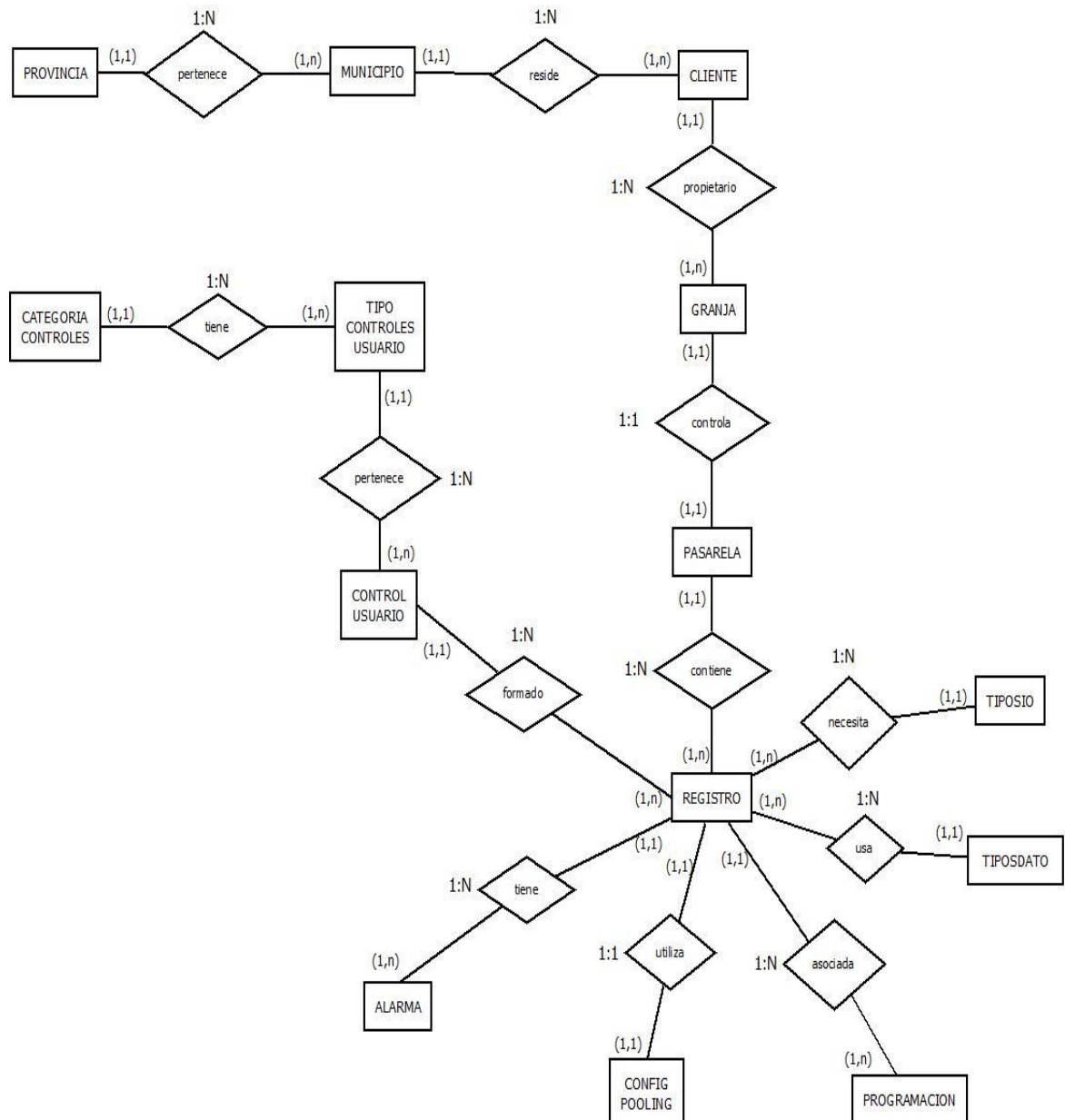


Imagen 59: Diagrama Entidad-Relación del sistema.

2. Tablas del sistema

A partir del diagrama Entidad-Relación se obtienen las siguientes tablas con sus atributos. Todas las tablas se encuentran en quinta forma normal, ya que se encuentran en cuarta forma normal y además cada relación de dependencia se encuentra definida por las claves candidatas.

Provincia: contiene los datos de las provincias en las que hay granjas.			
Nombre	Descripción	Tipo de dato	Clave Primaria
IdProvincia	Identificador de la provincia	uniqueidentifier	Si
Nombre	Nombre de la provincia	nvarchar(255)	No
ModifiedBy	Quien ha modificado el campo	nvarchar(255)	No
ModifiedOn	Cuando ha sido modificado el campo	smalldatetime	No
CreatedBy	Quien ha creado el campo	nvarchar(255)	No
CreatedOn	Cuando ha sido creado el campo	smalldatetime	No
Inactivo	Campo para no borrar los elementos pero dejarlos desactivados	bit	No

Municipio: contiene los datos de los municipios en los que hay granjas.			
Nombre	Descripción	Tipo de dato	Clave Primaria
IdMunicipio	Identificador del municipio	uniqueidentifier	Si
IdProvincia	Identificador de la provincia a la que pertenece el municipio	uniqueidentifier	No (FK)
Nombre	Nombre de la provincia	nvarchar(255)	No
ModifiedBy	Quien ha modificado el campo	nvarchar(255)	No
ModifiedOn	Cuando ha sido modificado el campo	smalldatetime	No
CreatedBy	Quien ha creado el campo	nvarchar(255)	No
CreatedOn	Cuando ha sido creado el campo	smalldatetime	No
Inactivo	Campo para no borrar los elementos pero dejarlos desactivados	bit	No

Cliente: contiene los datos de los clientes que poseen granjas.			
Nombre	Descripción	Tipo de dato	Clave Primaria
IdCliente	Identificador del cliente	uniqueidentifier	Si
IdMunicipio	Identificador del municipio al que pertenece el cliente	uniqueidentifier	No (FK)
CIF	CIF del cliente	nvarchar(14)	No
RazonSocial	Razón social del cliente	nvarchar(255)	No
Direccion	Dirección del cliente	nvarchar(255)	No
CP	Código postal del cliente	nvarchar(5)	No
Telefono1	Primer teléfono de contacto del cliente	nvarchar(15)	No
Telefono2	Segundo teléfono de contacto del cliente	nvarchar(15)	No

Fax	Fax de contacto del cliente	nvarchar(15)	No
Email	Email de contacto del cliente	nvarchar(50)	No
Observaciones	Observaciones a cerca del cliente	nvarchar(1000)	No
ModifiedBy	Quien ha modificado el campo	nvarchar(255)	No
ModifiedOn	Cuando ha sido modificado el campo	smalldatetime	No
CreatedBy	Quien ha creado el campo	nvarchar(255)	No
CreatedOn	Cuando ha sido creado el campo	smalldatetime	No
Inactivo	Campo para no borrar los elementos pero dejarlos desactivados	bit	No

Granja: contiene los datos de las granjas del sistema.			
Nombre	Descripción	Tipo de dato	Clave Primaria
IdGranja	Identificador de la granjas	uniqueidentifier	Si
IdCliente	Identificador del cliente que es dueño de la granja	uniqueidentifier	No (FK)
RefCatastral	Referencia catastral de la granja	nvarchar(50)	No
Nombre	Nombre de la granja	nvarchar(50)	No
Descripcion	Descripción de la granja	nvarchar(1000)	No
Polígono	Polígono en el que se encuentra la granja	nvarchar(50)	No
Parcela	Parcela en la que se encuentra la granja	nvarchar(50)	No
CentroParcela	Ubicación geográfica del centro de la granja	geography	No
ContornoParcela	Ubicación geográfica del contorno de la granja	geography	No
Superficie	Superficie que ocupa la granja (en m2)	int	No
ModifiedBy	Quien ha modificado el campo	nvarchar(255)	No
ModifiedOn	Cuando ha sido modificado el campo	smalldatetime	No
CreatedBy	Quien ha creado el campo	nvarchar(255)	No
CreatedOn	Cuando ha sido creado el campo	smalldatetime	No
Inactivo	Campo para no borrar los elementos pero dejarlos desactivados	bit	No

Pasarela: contiene los datos de las pasarelas que hay instaladas en las granjas. Las pasarelas son los sistemas que controlan los aparatos instalados en las granjas.			
Nombre	Descripción	Tipo de dato	Clave Primaria
IdPasarela	Identificador de la pasarela	uniqueidentifier	Si
IdGranja	Identificador de la granja en la que está instalada la pasarela	uniqueidentifier	No (FK)
Nombre	Nombre de la pasarela	nvarchar(255)	No
IP	IP para comunicación con la pasarela	nvarchar(15)	No
IdSlave	Identificador de esclavo de la pasarela en el sistema.	int	No
ModifiedBy	Quien ha modificado el campo	nvarchar(255)	No
ModifiedOn	Cuando ha sido modificado el campo	smalldatetime	No
CreatedBy	Quien ha creado el campo	nvarchar(255)	No

CreatedOn	Cuando ha sido creado el campo	smalldatetime	No
Inactivo	Campo para no borrar los elementos pero dejarlos desactivados	bit	No

Registro: contiene los datos de los registros que están conectados a las diferentes pasarelas del sistema.

Nombre	Descripción	Tipo de dato	Clave Primaria
IdRegistro	Identificador del registro	uniqueidentifier	Si
IdPasarela	Identificador de la pasarela	uniqueidentifier	No (FK)
Nombre	Nombre del registro	nvarchar(50)	No
Descripcion	Descripción del registro	nvarchar(255)	No
Direccion	Dirección del registro dentro de la pasarela	int	No
IdControl	Identificador del control de usuario al que pertenece	uniqueidentifier	No (FK)
TipoDato	Identificador del tipo de dato al que pertenece el registro	int	No (FK)
TipoIO	Identificador del tipo al que pertenece el registro	int	No (FK)
MinAnalogValue	Valor analógico mínimo que ha recibido el registro. Este valor puede ser nulo.	real	No
MaxAnalogValue	Valor analógico máximo que ha recibido el registro. Este valor puede ser nulo.	real	No
TipoDatoAnalogico	Indica que tipo de dato analógico contiene el registro.	nvarchar(5)	No
ModifiedBy	Quien ha modificado el campo	nvarchar(255)	No
ModifiedOn	Cuando ha sido modificado el campo	smalldatetime	No
CreatedBy	Quien ha creado el campo	nvarchar(255)	No
CreatedOn	Cuando ha sido creado el campo	smalldatetime	No
Inactivo	Campo para no borrar los elementos pero dejarlos desactivados	bit	No

TiposIO: contiene los datos de los distintos tipos de entrada o salida a los que pertenecen los registros.

Nombre	Descripción	Tipo de dato	Clave Primaria
TipoIO	Identificador del tipo	int	Si
Nombre	Nombre del tipo	nvarchar(50)	No
Descripcion	Descripción del tipo	nvarchar(100)	No
Inactivo	Campo para no borrar los elementos pero dejarlos desactivados	bit	No

TiposDato: contiene los datos de los distintos tipos de datos que pueden contener los registros.

Nombre	Descripción	Tipo de dato	Clave Primaria
TipoDato	Identificador del tipo de dato	int	Si
Nombre	Nombre del tipo de dato	nvarchar(50)	No

Descripcion	Descripción del tipo de dato	nvarchar(100)	No
Inactivo	Campo para no borrar los elementos pero dejarlos desactivados	bit	No

ControlUsuario: contiene los datos de los controles de usuario que se encargan de manejar los aparatos domóticos instalados en las granjas. Los controles de usuario están formados por registros.

Nombre	Descripción	Tipo de dato	Clave Primaria
IdControl	Identificador del control de usuario	uniqueidentifier	Si
IdPasarela	Identificador de la pasarela	uniqueidentifier	No (FK)
IdTipoControl	Identificador del tipo de control de usuario	uniqueidentifier	No (FK)
Nombre	Nombre del control de usuario	nvarchar(50)	No
Descripcion	Descripción del control de usuario	nvarchar(150)	No
Rectangulo	Rectángulo principal que ocupa el control de usuario. Se utiliza para posicionar correctamente el control de usuario en el grid	nvarchar(50)	No
Fila	Fila en la que se encuentra posicionado en la pantalla el control de usuario	int	No
Columna	Columna en la que se encuentra posicionado en la pantalla el control de usuario	int	No
Inactivo	Campo para no borrar los elementos pero dejarlos desactivados	bit	No

TipoControlesUsuario: contiene los datos de los tipos de controles de usuario a partir de los cuales se pueden crear los diferentes controles de usuario.

Nombre	Descripción	Tipo de dato	Clave Primaria
IdTipoControl	Identificador del tipo de control de usuario	uniqueidentifier	Si
IdCategoria	Identificador de la categoría a la que pertenece el control de usuario (hay diferentes tipos de controles de usuario)	int	No (FK)
Nombre	Nombre del control de usuario	nvarchar(50)	No
Descripcion	Descripción del control de usuario	nvarchar(255)	No
Icono	Ubicación del icono que representa el control de usuario. Se utiliza para representar el control de usuario en la barra de controles de usuario	nvarchar(100)	No
ImagenMostrar	Ubicación de la imagen que representa al control de usuario en la pantalla de configuración de la granja.	nvarchar(100)	No
RowSpan	Número de filas que ocupa el control de usuario. Se utiliza para posicionarlo correctamente en el grid de la pantalla.	int	No
ColumnSpan	Número de columnas que ocupa el control de usuario. Se utiliza para posicionarlo correctamente en el grid de la pantalla.	int	No
Programador	Indica el nombre del programador de tareas que necesita el control de usuario, ya que cada programador no tiene las	nvarchar(100)	No

	mismas características.		
Inactivo	Campo para no borrar los elementos pero dejarlos desactivados	bit	No

CategoriaControles: contiene los datos de las distintas categorías a las que pertenecen los controles de usuario.			
Nombre	Descripción	Tipo de dato	Clave Primaria
IdCategoria	Identificador de la categoría de control de usuario.	Int	Si
TipoControl	Nombre del tipo de control	nvarchar(50)	No

Alarma: contiene los datos de las alarmas que hay programadas para los registros de los controles de usuario.			
Nombre	Descripción	Tipo de dato	Clave Primaria
IdAlarma	Identificador de la alarma	uniqueidentifier	Si
IdRegistro	Identificador del registro al que afecta	uniqueidentifier	No (FK)
Nombre	Nombre de la alarma	nvarchar(255)	No
ValorAlarma	Valor al que se ha establecido la alarma (convertido al valor válido para el registro al que pertenece).	int	No
IdTipoAlarma	Identificador del tipo de alarma	int	No (FK)
ValorMostrar	Valor al que se ha establecido la alarma (valor entendible para el usuario)	real	No
Tolerancia	Tolerancia que acepta la alarma antes de activarse	int	No
Operativa	Indica si la alarma se encuentra en funcionamiento.	bit	No
ModifiedBy	Quien ha modificado el campo	nvarchar(255)	No
ModifiedOn	Cuando ha sido modificado el campo	smalldatetime	No
CreatedBy	Quien ha creado el campo	nvarchar(255)	No
CreatedOn	Cuando ha sido creado el campo	smalldatetime	No
Inactivo	Campo para no borrar los elementos pero dejarlos desactivados	bit	No

TipoAlarma: contiene los datos de los diferentes tipos de alarmas que se pueden establecer.			
Nombre	Descripción	Tipo de dato	Clave Primaria
IdTipoAlarma	Identificador del tipo de alarma	int	Si
Descripcion	Descripción del tipo de alarma	nvarchar(50)	No
Inactivo	Campo para no borrar los elementos pero dejarlos desactivados	bit	No

HistoricoAlarma: contiene los históricos de las alarmas que han sucedido en las granjas.			
Nombre	Descripción	Tipo de dato	Clave Primaria

IdHistoricoAlarma	Identificador de cada entrada del histórico de alarmas	int	Si
IdAlarma	Identificador de la alarma	uniqueidentifier	No (FK)
TimeStamp	Momento en el que ha sucedido la alarma	Smalldatetime	No
Activado	Indica si la entrada del histórico esta activa o ha sido ya comprobada.	bit	No

ConfigPooling: contiene los datos de las configuraciones de pooling de los registros que se encuentran instalados en las granjas.

Nombre	Descripción	Tipo de dato	Clave Primaria
IdPooling	Identificador de la configuración de pooling	uniqueidentifier	Si
IdRegistro	Identificador del registro al que afecta	uniqueidentifier	No (FK)
Intervalo	Valor que indica cada cuanto se debe de comprobar el valor del registro	int	No
UltimaEjecucion	Momento en el que se ha realizado el pooling por última vez	smalldatetime	No
ProximaEjecucion	Momento en el que se debe de realizar el pooling la siguiente vez	smalldatetime	No
Estado	Indica si el registro se encuentra en un estado de alarma.	bit	No
Operativo	Indica si el pooling se encuentra en funcionamiento	bit	No
Activado	Indica si la entrada del histórico esta activa o ha sido ya comprobada.	bit	No

HistoricoRegistro: contiene los históricos de los valores que han tomado los registros.

Nombre	Descripción	Tipo de dato	Clave Primaria
IdLog	Identificador de las entradas del histórico	Uniqueidentifier	Si
IdPooling	Identificador de la configuración de pooling	uniqueidentifier	No (FK)
fechaRecogida	Momento en el que ha sido recogido el valor	smalldatetime	No
ValorRegistro	Valor que se ha recogido del registro	int	No
ValorMostrar	Valor recogido del registro una vez que se ha convertido a los valores reales.	real	No
EsAcumulado	Indica si el valor obtenido del registro es acumulado o si es un valor nuevo	bit	No

Programacion: contiene los datos de las programaciones que se han establecido en los registros que componen los controles de usuario.

Nombre	Descripción	Tipo de dato	Clave Primaria
IdTarea	Identificador de la tarea	Uniqueidentifier	Si
IdRegistro	Identificador del registro al que afecta	uniqueidentifier	No (FK)
Nombre	Nombre de la tarea programada	nvarchar(50)	No
Descripcion	Descripción de la tarea programada	nvarchar(100)	No
IdDesencadenador	Identificador del desencadenar que provoca que se ejecute la tarea	uniqueidentifier	No (FK)

FechaInicio	Fecha de inicio de la tarea	date	No
FechaFin	Fecha de fin de la tarea (con hora incluida)	time(7)	No
RepetirDias	Cada cuantos días se quiere repetir la tarea	int	No
DiasSemana	Días de la semana en los que se quiere ejecutar la tarea	nvarchar(55)	No
Operativo	Indica si la tarea se encuentra activa	bit	No
ModifiedBy	Quien ha modificado el campo	nvarchar(255)	No
ModifiedOn	Cuando ha sido modificado el campo	smalldatetime	No
CreatedBy	Quien ha creado el campo	nvarchar(255)	No
CreatedOn	Cuando ha sido creado el campo	smalldatetime	No
Inactivo	Campo para no borrar los elementos pero dejarlos desactivados	bit	No

TipoProgramacion: contiene los datos de los distintos tipos de programación que se pueden crear.			
Nombre	Descripción	Tipo de dato	Clave Primaria
IdDesencadenador	Identificador del tipo de programación.	uniqueidentifier	Si
Nombre	Nombre del tipo de programación	nvarchar(50)	No
Descripcion	Descripción del tipo de programación	nvarchar(100)	No
Inactivo	Campo para no borrar los elementos pero dejarlos desactivados	bit	No

3.1.3 Diseño de la capa de dominio

La capa de dominio va a ser la encargada de contener toda la funcionalidad de la aplicación. Aquí deberemos de definir todas las clases del sistema con sus correspondientes funciones, las cuales nos permitirán completar todas las acciones que el sistema necesite.

3.1.3.1 Diagrama de clases de diseño

A continuación, se van a mostrar las clases de diseño del proyecto divididas en 4 módulos de trabajo para así facilitar el desarrollo de la aplicación y realizar una aplicación más clara y simple.

- **Módulo InstaladorGranja**

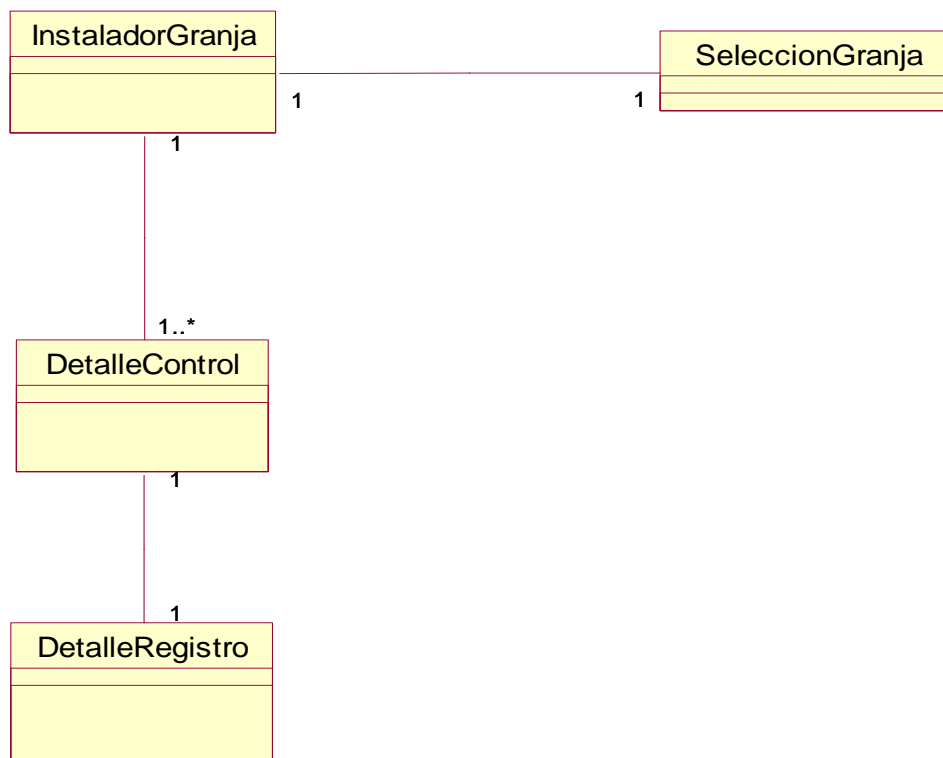


Imagen 60: Diseño de clases del módulo InstaladorGranja.

Este módulo tiene como función principal facilitar la introducción de datos al sistema por parte del instalador domótico. Así, cuando el instalador acuda a una granja a instalar los aparatos, mediante este módulo va a poder introducir, para esa granja en concreto, los aparatos que ha instalado y sus características técnicas. A partir de estos datos, podrán funcionar el resto de los módulos del sistema.

- **Módulo Telecentro**



Imagen 61: Diseño de clases del módulo Telecentro.

Este módulo se va a encargar únicamente de mostrar todas las granjas que se encuentran dadas de alta en el sistema indicando el estado en el que se encuentran. De tal modo que cuando algún aparato de una granja no funcione adecuadamente o muestre valores anómalos, esta pantalla mostrará un indicador avisando de tal circunstancia.

Mediante esta pantalla, se tendrán monitorizadas en todo momento todas las granjas del sistema, mostrando un indicador del estado general de la granja. También se podrá visualizar, de manera más detallada, el estado de cada uno de los aparatos de las granjas, para así poder detectar rápidamente donde está ocurriendo el error.

- **Módulo PantallaConfiguración**

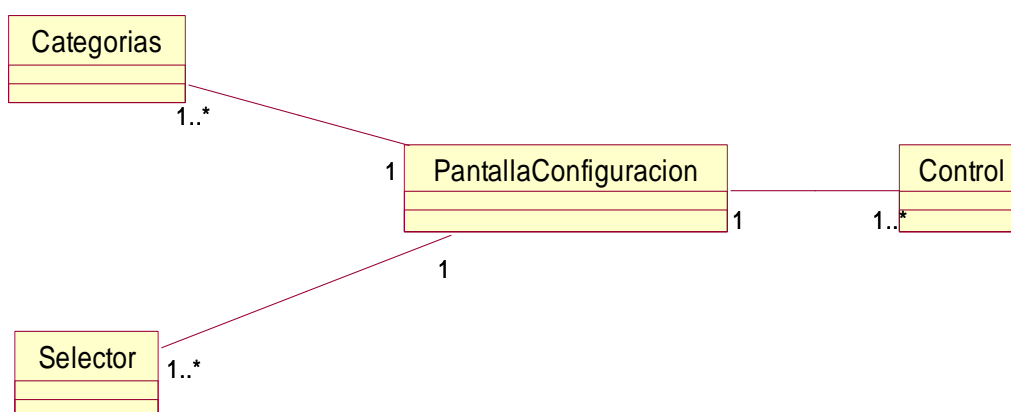


Imagen 62: Diseño de clases del módulo PantallaConfiguración.

Este módulo es el responsable de la organización de los controles de usuario en la pantalla. Mediante este módulo se deben de poder arrastrar los controles de usuario a la pantalla hasta situarlos en un punto concreto. Una vez guardada la configuración, esta será la que se utilice para posicionar los controles en la pantalla de visualización de la granja.

Está formado por un selector, que hace posible ordenar por categorías y de manera dinámica los diferentes controles que tiene instalados la granja seleccionada. Como cada granja tiene distintos controles de usuario, es necesario que la obtención y ordenación de los controles sea totalmente independiente de cuantos tipos de controles haya en la granja.

- **Módulo VisualizaciónGranja**

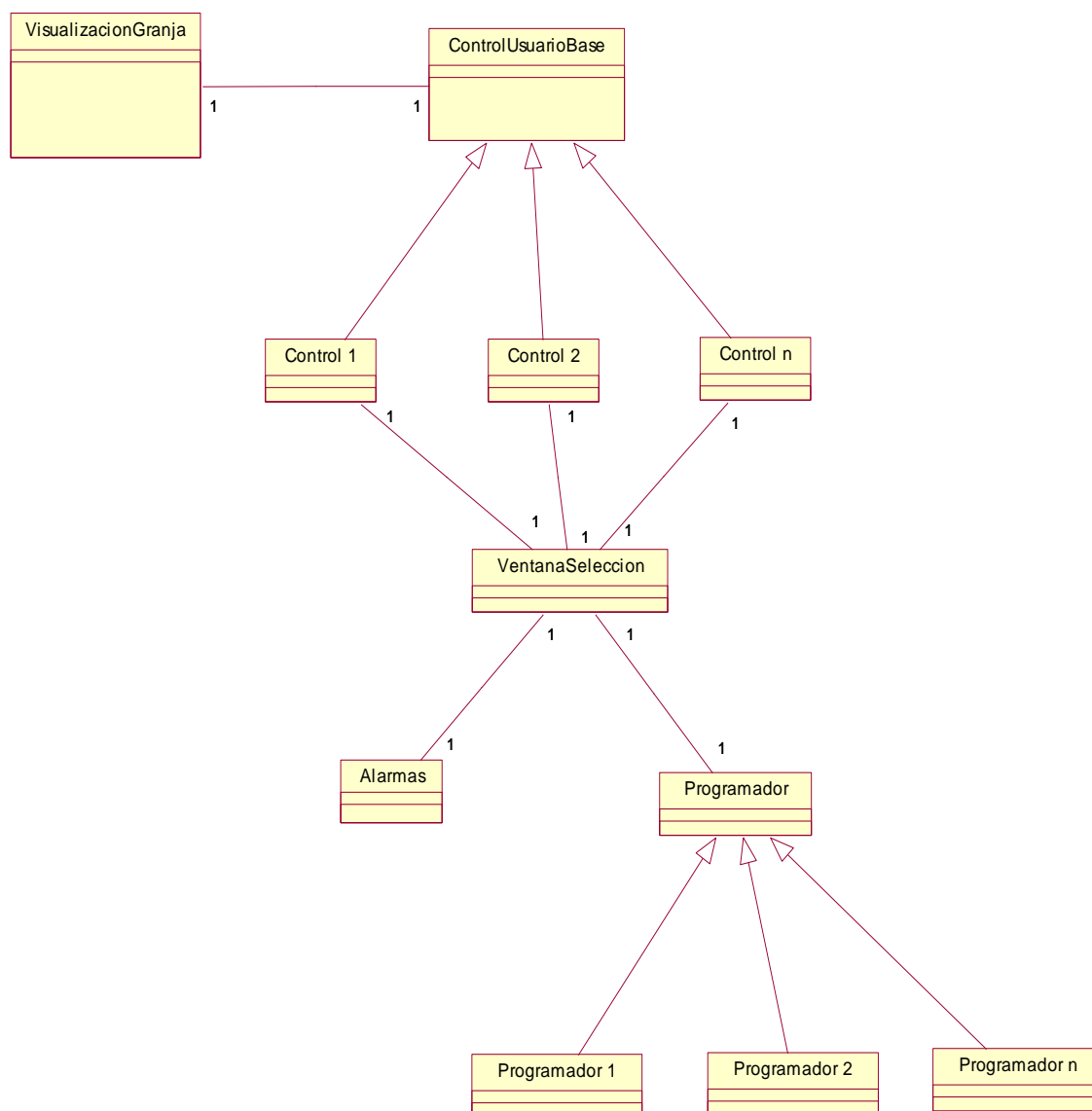


Imagen 63: Diseño de clases del módulo VisualizacionGranja.

Este módulo se va a encargar del manejo de los diferentes aparatos que se encuentran instalados en la granja que se ha seleccionado previamente. Para ello tendremos un control de usuario por cada aparato que se encuentre en la granja. Mediante este control de usuario seremos capaces de manejar el aparato y de establecerle alarmas y programaciones según nuestras necesidades.

Por ejemplo, tenemos un foco halógeno instalado en la granja. Mediante un control de usuario específico para focos halógenos seremos capaces de encenderlo y apagarlo así como de establecer la intensidad de la luz. Además podremos establecer que el foco se encienda, por ejemplo a las 11:00 PM. Y también podremos establecer una alarma que nos indique si el foco se encuentra encendido más tarde de las 11:30 PM.

Como no se sabe cuántos tipos de controles de usuario va a haber en el sistema se debe de crear este módulo de la forma más abstracta posible. Así debería de darnos igual poner en la pantalla un control de usuario para una fluorescente que para un ventilador. Esto es algo muy importante, ya que si en un futuro se añaden nuevos controles de usuario no será necesario modificar la programación de este módulo para que puedan cargarse los controles de usuario nuevos.

3.1.3.1 Funciones principales de la aplicación

A continuación voy a detallar las diferentes funciones de cada uno de los módulos de la aplicación para así comprender mejor su funcionamiento.

- **Módulo InstaladorGranja**

- Pantalla Principal

- cargarGranjas(): función que se encarga de obtener los datos de las granjas desde la base de datos.
- cargarControles(): función que se encarga de obtener los datos de los controles de usuario desde la base de datos.
- cargarControles_Completed(): evento que ocurre al obtener todos los datos de los controles. En ese momento asignamos los datos a un grid para mostrarlos por pantalla.
- cargarGranjas_Completed(): evento que ocurre al obtener todos los datos de las granjas. En ese momento asignamos los datos a un grid para mostrarlos por pantalla.
- btnNuevo_Click(): al pulsar el botón abrimos la ventana para dar de alta un nuevo control de usuario.
- btnModificar_Click(): al pulsar el botón abrimos la ventana para modificar un control de usuario previamente seleccionado.
- btnEliminar_Click(): al pulsar el botón eliminamos el control de usuario seleccionado previamente.
- configurarGrid(): función para configurar de manera dinámica el grid que muestra los datos por pantalla.

- SelecciónGranja

- ChildWindow_Loaded(): evento que se produce al ser cargada la ventana emergente. Cuando esto ocurre debemos de asignar los datos de las granjas al control combobox que tiene la ventana emergente.
- OKButton_Click(): al pulsar el botón debemos de cerrar la ventana emergente con la opción true.
- CancelButton_Click(): al pulsar el botón debemos de cerrar la ventana emergente con la opción false.

- DetalleControl

- cargarControl(): función que se encarga de obtener los datos del control de usuario desde la base de datos.
- cargarRegistros(): función que se encarga de obtener los datos de los registros vinculados al control de usuario desde la base de datos.
- cargarCombo(): función que se encarga de obtener los distintos tipos de controles de usuario desde la base de datos.

- cargarRegistros_Completed(): evento que ocurre al cargarse completamente los datos desde la base de datos. Cuando esto ocurre asignamos los datos a un grid para mostrarlos por pantalla.
- cargarControl_Completed(): evento que ocurre al cargarse completamente los datos desde la base de datos. Cuando esto ocurre asignamos los valores del control de usuario a los cuadros de texto.
- cargarCombo_Completed():evento que ocurre al cargarse completamente los datos desde la base de datos. Cuando esto ocurre asignamos los valores obtenidos al control combobox que los mostrará en pantalla.
- btnNuevo_Click(): al pulsar el botón abrimos la pantalla para introducir un nuevo registro.
- btnModificar_Click(): al pulsar el botón abrimos la pantalla para modificar un registro vinculado a un control de usuario previamente seleccionado.
- btnEliminar_Click(): al pulsar el botón eliminamos el registro previamente seleccionado.
- btnGuardar_Click(): al pulsar el botón guardamos los cambios realizados y los dejamos de forma definitiva en la base de datos.
- camposCorrectos(): comprobamos que no se han dejado ningún campo sin introducir.
- DetalleRegistro
 - cmbTipoDato_SelectionChanged(): el evento SelectionChanged ocurre cuando un control combobox cambia su elemento seleccionado. Cuando esto ocurre debemos de comprobar cuál es la selección actual para mostrar unos valores ocultos. Como este evento hay 7 más que realizan la misma función.
 - camposCorrectos(): función que comprueba que se han introducido todos los campos antes de proceder a guardar los datos.

- **Módulo InstaladorGranja**

- Pantalla Principal
 - OnDragQuery(): evento del grid que ocurre al seleccionar un elemento que se encuentra dentro del grid.
 - OnDragInfo(): evento del grid que ocurre al seleccionar e intentar arrastrar un elemento que se encuentra en el grid.
 - OnDropQuery(): evento que ocurre al intentar posicionar un elemento dentro del grid.
 - OnDropInfo(): evento que ocurre al dejar un elemento en un punto dentro del grid.
 - btnGuardar_Click: evento que ocurre al pulsar el botón guardar y por el cual se guardan las posiciones de los controles de usuario en la pantalla. Se accede a la base de datos para dejar los datos guardados.

- `asignarRectángulos()`: para posicionar los controles correctamente se debe de dividir el grid en rectángulos que se numeran mediante esta función.
- `bloquearRectángulos()`: para evitar se superpongan los controles arrastrados al grid se deben de bloquear los rectángulos que ocupan estos controles.
- `desbloquearRectángulos()`: al mover un control de usuario a otra posición se deben de desbloquear los rectángulos que ocupaba antes para así poder posicionar en ellos otro control de usuario.
- `posicionCorrecta()`: función que indica si un control de usuario se puede posicionar correctamente sin salirse de los márgenes de la pantalla.
- **Control**
 - `establecerImagen()`: función que establece la imagen que representa al control de usuario.
 - `btnConfig_Click()`: al pulsar el botón de configuración se abre la ventana en la que se selecciona un registro del control de usuario.
 - `btnProgramador_Click()`: al pulsar el botón se abre un programador específico según el tipo de control de usuario en el que estemos.
 - `brnInfo_Click()`: nos muestra información acerca del control de usuario en el que estamos.
- **VentanaSeleccion**
 - `configurarGrid()`: función encargada de configurar el grid que va a mostrar los datos por pantalla.
 - `cargarRegistrosAlarma()`: función que se encarga de obtener los registros que contienen alarma desde la base de datos.
 - `cargarRegistrosProgramador()`: función que se encarga de obtener los registros que contienen programador de tareas desde la base de datos.
 - `btnAlarma_Click()`: abrimos la pantalla de configuración de alarmas para el registro seleccionado.
 - `btnTarea_click()`: abrimos la pantalla de configuración de tareas para el registro seleccionado. Según el tipo de control de usuario que sea cargará un configurador distinto.
- **VentanaAlarmas**
 - `configurarGrid()`: función que configura el grid que va a mostrar los datos por pantalla.
 - `cargarAlarmas()`: función que carga las alarmas para un registro previamente seleccionado, obteniendo los datos desde base de datos.
 - `cargarTipoAlarmas()`: función que obtiene los tipos de alarmas desde la base de datos.
 - `buscarAlarma()`: función que se encarga de buscar una alarma dentro del conjunto de alarmas.
- **VentanaNuevaAlarma**

- `btnGuardar_Click()`: al pulsar el botón de guardar se guardan los datos introducidos en la base de datos.
- `camposCorrectos()`: función que comprueba que se han introducido todos los campos necesarios.

- **Módulo Telecentro**

- **Página Principal**

- `cargarIntervalos()`: función que se encarga de obtener desde la base de datos los valores de los intervalos de refresco de cada elemento de las granjas.
- `cargarControles()`: función que carga los datos de los controles de usuario desde la base de datos.
- `cargarRegistrosRefresco()`: función que se encarga de obtener desde la base de datos los valores de los registros en cada intervalo de refresco.
- `SetUpTimer()`: función que configura el timer de la página para realizar los refrescos.
- `Timer1_tick`: evento que ocurre al tiempo configurado al timer. Cuando ocurre se deben de obtener los datos de los registros para obtener sus nuevos valores.
- `TimerNotificaciones()`: timer para mostrar las notificaciones emergentes cuando se produce alguna alarma en alguna granja.
- `Botón_Click()`: cuando pulsamos el botón hacia la derecha de una granja se deben de desplegar una serie de indicadores que nos marcan el estado de los aparatos conectados a la granja.
- `botonIzq_Click()`: cuando pulsamos el botón hacia la izquierda de una granja de deben de ocultar los indicadores que nos marcan el estado de los aparatos conectados a la granja.
- `buscarStackGranja()`: función que busca el elemento que contiene a una granja dentro de la pantalla. Se necesita para realizar los cambios en los estados cuando se produce el refresco.
- `notificarAlarma()`: al producirse una alarma en algún aparato de una granja se muestra una ventana emergente abajo a la derecha de la pantalla. Esta función se encarga de crear esta notificación y mostrarla.
- `notificarError()`: al producirse algún error en las comunicaciones se debe de mostrar una ventana emergente abajo a la derecha de la pantalla. Esta función se encarga de crear esta notificación y mostrarla.

- **IndicadorEstado**

- `cambiarIndicador()`: cambia la visualización del indicador de alarma según se esté o no produciendo una alarma.

- `setToolTip()`: función para establecer un tooltip al indicador de estado, de manera que cuando pasemos el ratón por encima sepamos a que control representa ese indicador.
 - `actualizarTooltip()`: función que actualiza el contenido del tooltip cuando se refrescan los datos.
-
- **Módulo VisualizaciónGranja**
 - Página principal
 - `cargarControles()`: función que obtiene los datos desde la base de datos de los controles de usuario que se deben de cargar para la granja seleccionada.
 - `cargarControles_Completed()`: cuando ya se tienen todos los datos de los controles que hay que cargar, debemos de crear un control de usuario genérico con los datos obtenidos. De este modo nos es independiente que tipo de control de usuario estemos cargando y si en un futuro añadimos nuevos controles de usuario no deberemos de cambiar la programación.

3.1.4 Diseño de la capa de presentación

En esta capa se van a realizar todas las interfaces de usuario del sistema, es decir, esta capa es la capa visible de la aplicación y con la que tendrá contacto el usuario final del sistema. Por ello, se deben de cumplir una serie de pautas estéticas, para que el uso de la aplicación por parte del usuario final sea lo más sencilla y amigable posible.

En mi caso, las interfaces carecen del diseño y de las características propias de un producto final, ya que en la fase en la que dejé el proyecto en la empresa aun no se pedía una interfaz de usuario amigable.

A continuación voy a mostrar las distintas interfaces del sistema divididas por módulos. De cada una de ellas realizaré una pequeña explicación sobre sus partes y su función.

- **Módulo Instalador**
 - **SelecciónGranja**

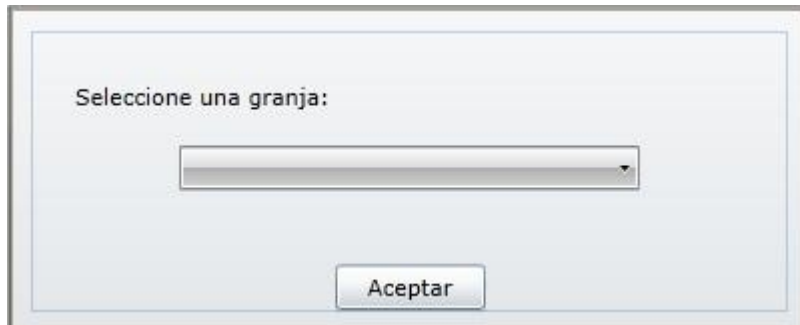


Imagen 64: Pantalla SelecciónGranja del módulo Instalador

Mediante esta pantalla, el instalador domótico selecciona una de las granjas del sistema. Este es el primer paso necesario antes de añadir o modificar cualquier parámetro de una granja.

En el menú desplegable se cargan todas las granjas activas del sistema, entonces el instalador únicamente debe de seleccionar una de las granjas y pulsar el botón aceptar.

○ PaginaPrincipal

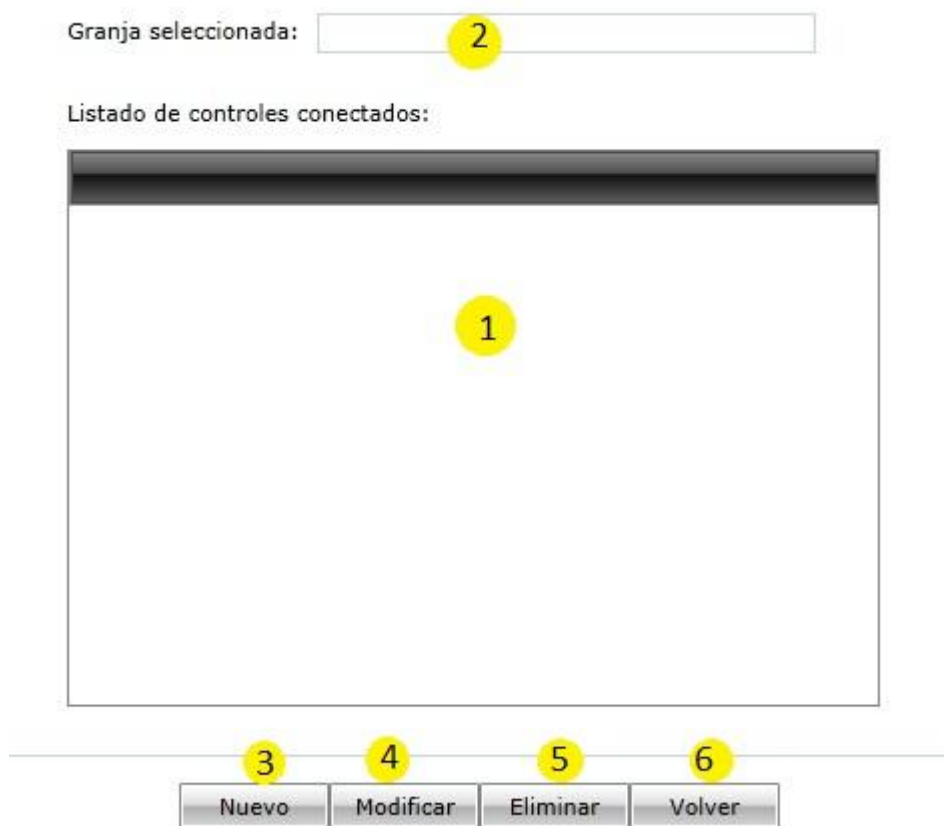


Imagen 65: Pantalla PáginaPrincipal del módulo Instalador

Esta es la pantalla principal del módulo Instalador. Se accede a ella después de haber seleccionado una granja en la pantalla SeleccionGranja. Esta es la pantalla que nos va a derivar a las distintas acciones que se pueden realizar desde el módulo Instalador. A continuación explico cada parte de la pantalla:

1. Listado (grid) en el que aparecerán los controles que ya se encuentran dados de alta para la granja. Si la granja es nueva la lista saldrá vacía.
2. Indicador de la granja en la que estamos ubicados.
3. Botón Nuevo: botón para dar de alta un nuevo control (se abre la ventana DetalleControl).
4. Botón Modificar: botón para modificar un control seleccionado de la lista de controles (se abre la ventana DetalleControl).
5. Botón Eliminar: botón para eliminar un control seleccionado de la lista de controles.
6. Botón Volver: botón para volver a la ventana anterior (SelecciónGranja).

- **DetalleControl**

Detalle de Control de Usuario

Nombre: **2** Guardar Control

1 Descripción:

Tipo Control:

Registros asociados al control de usuario:

3

4 Nuevo **5** Modificar **6** Eliminar **7** Volver

Imagen 66: Pantalla DetalleControl del módulo Instalador

Esta es la pantalla en la que se especifican los atributos principales de un control de usuario (nombre, descripción y tipo de control), así como los registros que tiene asociados. A continuación explico cada parte de la pantalla:

1. Datos del control: En esta parte se deben de introducir los datos generales del control de usuario.
2. Botón Guardar Control: con este botón se guardan en la base de datos los datos generales del control de usuario que se han introducido. Se realiza primero una validación de los datos para asegurarse que no se queda ningún campo sin rellenar o que los datos son incorrectos.
3. Listado (grid) en el que se muestran los registros que forman parte del control de usuario (ya que un control de usuario puede estar formado por uno o más registros domóticos).
4. Botón Nuevo: con este botón se crea un nuevo registro que va a estar relacionado con el control de usuario actual (se abre la ventana DetalleRegistro).
5. Botón Modificar: con este botón se modifica un registro previamente seleccionado de la lista de registros (se abre la ventana DetalleRegistro).
6. Botón Eliminar: con este botón se eliminar un registro previamente seleccionado de la lista de registros.
7. Botón Volver: con este botón se vuelve a la ventana anterior (PaginaPrincipal).

- **DetalleRegistro**

Datos del Registro

Código:

Nombre:

Descripción:

Dirección:

Tipo IO: Tipo Dato:

Valor Mínimo: Valor Máximo: Tipo Dato Analógico:

Refresco Activo: ☐ Intervalo de refresco (minutos):

Imagen 67: Pantalla DetalleRegistro del módulo Instalador

Esta es la pantalla en la que se especifican los atributos de un registro. Estos parámetros son técnicos y los debe de completar el técnico que ha realizado la instalación de la granja. Estos valores dependerán de registro que se esté introduciendo, así que hay campos que se muestran o se ocultan dependiendo de la selección que se haga en el campo TipoIO y en el campo TipoDato.

Una vez introducidos todos los datos, se debe de pulsar el botón Guardar para que se guarden los datos en la base de datos. El botón Volver nos devolverá a la ventana DetalleControl.

- **Módulo PantallaConfiguracion**

- **PantallaPrincipal**

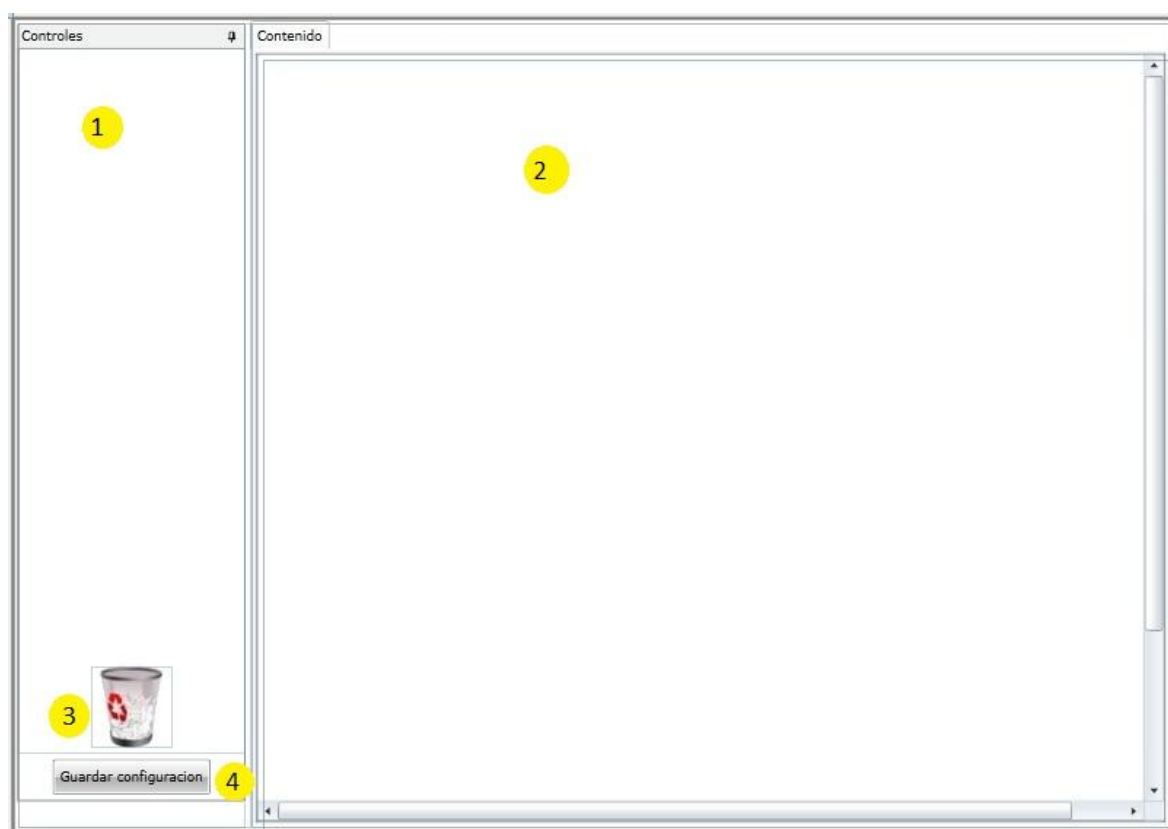


Imagen 68: Pantalla PaginaPrincipal del módulo PantallaConfiguracion

Esta es la pantalla principal del módulo PantallaConfiguración. Esta pantalla se utiliza para configurar la posición de los controles de usuario en la pantalla. Para ello, se deben de arrastrar desde una lista de controles (1) hasta la zona que representa la pantalla real (2). Los controles se podrán mover tantas veces como se quiera por la pantalla hasta encontrar la posición más adecuada a nuestras necesidades. Cuando ya tengamos esa posición deseada, será suficiente con pulsar el botón Guardar Configuración (4) y los datos de las posiciones de los controles quedarán almacenados en la base de datos.

Si en algún momento queremos quitar alguno de los controles que tenemos en la pantalla, bastará con arrastrarlo desde su posición hasta la Papelera (3). En ese momento, el control dejará de estar visible en la pantalla y volverá a colocarse en la lista de controles, ordenador según su categoría. Nuevamente habrá que pulsar el botón de guardar para hacer que los cambios tengan efecto.

- **Control de Usuario**



Imagen 69: Control de usuario de la pantalla PaginaPrincipal

Cuando en la pantalla PaginaPrincipal arrastramos un control desde la lista de controles y lo dejamos en zona de representación de la pantalla, aparece una imagen (1) que representa la imagen real del control de usuario que hemos arrastrado. En realidad es una imagen fija de lo que es en sí el control (si arrastramos un interruptor veremos la imagen de un interruptor, pero no podremos accionarlo). Esta imagen tendrá el mismo tamaño que el control de usuario al que representa, para así poder ubicar correctamente todos los controles en la pantalla.

Además, al pasar con el ratón por encima de la imagen aparecerán, en el extremo superior derecho, una serie de botones (2, 3, 4). El número de estos botones dependerá de los registros que tenga vinculados el control de usuario. El primer botón (2) se utiliza para abrir la ventana de configuración de alarmas (VentanaAlarmas), el segundo botón (3) se utiliza para abrir la ventana del programador de tareas (Programador) y el tercer botón (4) aparece siempre ya que pulsándolo nos aparecen las características del control de usuario en el que estamos.

Este objeto es el mismo para todos los controles de usuario, variando sólo en la imagen que se muestra según el control de usuario que representa.

- **VentanaSelección**



Imagen 70: Pantalla VentanaSeleccion del módulo PantallaConfiguracion

En esta pantalla se selecciona el registro de la lista (1) al que se quiere agregar una alarma o una programación. No todos los registros admiten alarmas o programaciones, todo depende del tipo de registro del que se trate. Por tanto, según el tipo de registro que se seleccione, se mostrarán u ocultarán los botones Crear Alarma (2) y Crear Tarea (3). Para volver a la PantallaPrincipal se deberá de pulsar el botón volver (4).

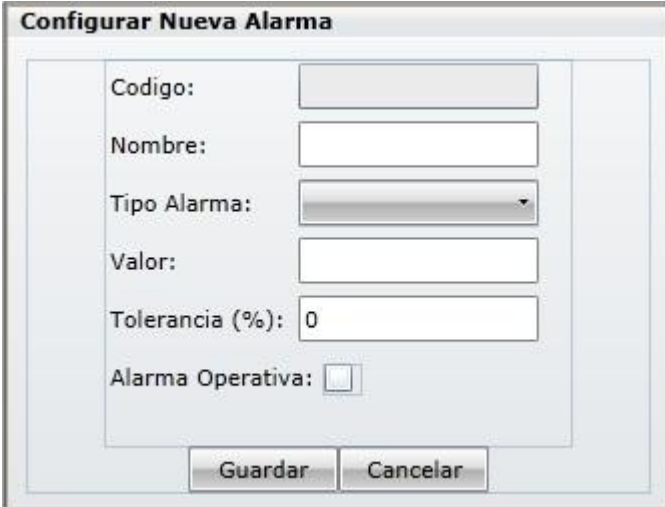
- **VentanaAlarmas**

Imagen 71: Pantalla VentanaAlarmas del módulo PantallaConfiguracion

Esta pantalla es en la que se dan de alta las alarmas para un registro previamente seleccionado. La pantalla está formada por los siguientes elementos:

1. Lista (grid) en el que se muestran las alarmas ya creadas para ese registro.
2. Parámetros necesarios para configurar la alarma correctamente.
3. Botón Eliminar: para eliminar una alarma seleccionada de la lista de alarmas.
4. Botón Nueva: para crear una nueva alarma para el registro (se abre la pantalla NuevaAlarma).
5. Botón Guardar: para guardar los cambios de la alarma seleccionada. Siempre se realiza una comprobación de los datos antes de guardarlos.
6. Botón Volver: para volver a la ventana anterior.

- **VentanaNuevaAlarma**



Configurar Nueva Alarma

Codigo:

Nombre:

Tipo Alarma:

Valor:

Tolerancia (%):

Alarma Operativa: ☐

Imagen 72: Pantalla VentanaNuevaAlarma del módulo PantallaConfiguracion

Esta es la pantalla con la que se introducen los parámetros necesarios para crear una nueva alarma. Para ello, basta con rellenar los campos de la pantalla y pulsar el botón Guardar para que los cambios se guarden en la base de datos. Antes de guardar los datos se realiza una comprobación de los mismos para comprobar que son correctos. Si se desea volver a la ventana anterior (VentanaAlarmas) habrá que pulsar el botón Cancelar.

- **Programador**

Imagen 73: Pantalla Programador del módulo PantallaConfiguracion

Este es el aspecto general que tiene la ventana del programador de tareas. Digo que es el aspecto general porque puede haber distintos programadores de tareas, pero todos heredan una serie de características de un programador general. Es necesario hacerlo así porque no todos los controles de usuario tienen las mismas características y, por tanto, es imposible crear un único programador para todos ellos.

Para crear una tarea habrá que rellenar los parámetros que se muestran por pantalla (1) y después pulsar el botón Guardar (4). Si por el contrario queremos eliminar la tarea en la que estamos situados, bastará con pulsar el botón Eliminar (2). Por último, si queremos crear una nueva tarea, tendremos que pulsar el botón Nueva (3) y automáticamente la pantalla se limpiará y podremos introducir los datos de la nueva tarea. Para volver a la ventana anterior (VentanaSeleccion) deberemos de pulsar el botón Volver (5).

- **Módulo Telecentro**
 - **PantallaPrincipal**



Imagen 74: Pantalla PantallaPrincipal del módulo Telecentro

Esta es la pantalla principal desde la cual se van a tener monitorizadas todas las granjas que se encuentren dadas de alta en el sistema. Para ello, se mostrarán en pantalla todas las granjas en una columna (1) y a su derecha aparecerán:

- Estado general de la granja (2): indica el estado global de la granja, si todas las alarmas están OK el estado de la granja será OK. En cuanto una alarma se active el estado de la granja cambiara a Alarma activada.
- Estado de cada alarma (3): indica el estado de cada alarma que hay programada en la granja. Al pasar por encima de cada indicador podremos ver a qué control se refiere la alarma.

Además, cada vez que ocurra una alarma, aparecerá una notificación en el área de notificaciones de la pantalla (4), que nos indicará en que granja a sucedido la alarma. Este sistema es importante, ya que si no entran en la pantalla todas las granjas puede que alguna tenga alguna alarma activada y no nos demos cuenta.

- **IndicadorEstado**



Imagen 75: Indicador de estado

Este objeto se va a encargar de mostrarnos el estado del control de usuario al que representa. Así, cada control de usuario al que se le haya programado una alarma contará con un control de este tipo en la PantallaPrincipal del módulo Telecentro. Además, al pasar el ratón por encima de este control, nos mostrará información acerca del control de usuario al que representa y de los valores de las alarmas que hay configuradas.

- **Módulo VisualizaciónGranja**

- **PantallaPrincipal**

De esta pantalla no hay una imagen en concreto, ya que depende de la granja en la que se esté ubicado. Cada granja va a tener unos controles de usuario distintos, pero todos ellos heredan unas características comunes de un control de usuario estándar. Así, podemos generalizar la forma de posicionar los controles de usuario en la pantalla, no importándonos qué tipo de control de usuario es.

Este hecho tiene una gran importancia, ya que si se necesita crear un nuevo control de usuario porque se ha instalado un nuevo aparato en una granja que aun no se había instalado en ninguna otra, no será necesario tocar nada del código de esta página y directamente se podrá cargar el control nuevo.

Hay que recordar que la posición en la que se cargan los controles de usuario en esta pantalla depende de cómo se hayan dispuesto en la pantalla PantallaConfiguración.

Capítulo 4

Conclusiones y líneas futuras

Para empezar, debo de decir que la realización de este proyecto ha supuesto para mí una gran motivación y un gran esfuerzo, debido a que se trata de un proyecto real que se está llevando a cabo entre un conjunto de empresas de Navarra, en una de las cuales he estado 6 meses con una beca de formación de la Universidad Pública de Navarra.

Por ello, el reto de integrarme en una empresa importante y poner en práctica todos mis conocimientos adquiridos durante mis estudios en la universidad ha supuesto un logro importante a nivel personal, así como una nueva forma de adquirir nuevos conocimientos gracias al trabajo en grupo realizado con los distintos trabajadores de la empresa.

El proyecto realizado está basado en una idea sencilla pero con grandes posibilidades de crecimiento. Se ha tratado de partir de un sistema simple (solo mostrar unas variables domóticas en un explorador web) para poco a poco ir aumentando la complejidad del sistema, hasta conseguir crear un sistema de control centralizado de las granjas del sistema donde poder observar el funcionamiento simultáneo de todas ellas.

Para el desarrollo de este sistema se ha utilizado la plataforma Silverlight 4. Este ha sido uno de los inconvenientes más importantes que se han tenido a la hora de desarrollar la aplicación, ya que la documentación existente sobre Silverlight 4 no es muy clara y no se encuentran muchas ayudas para el desarrollo de grandes aplicaciones con esta plataforma. Por ello, han sido necesarias muchas horas de documentación y de pruebas, así como reestructuraciones en el código para intentar conseguir una aplicación bien estructurada. Como punto más complicado, destacaré las comunicaciones con la base de datos, ya que se alejan del método tradicional de las conexiones a datos e introducen conceptos nuevos que no están muy bien explicados y que poca gente llega a dominar.

Como partes importantes de la aplicación destacaré las siguientes: la pantalla de configuración y la pantalla de visualización de la granja. En la pantalla de configuración hay que destacar la técnica Drag & Drop, por la cual se pueden arrastrar los controles por la pantalla para posicionarlos en la parte que más nos interesen. Del mismo modo se pueden arrastrar hasta una papelera para que no aparezcan en la pantalla. Esta técnica supuso bastantes horas de pruebas hasta conseguir que todo funcionase correctamente. También hay que destacar el menú lateral izquierdo, en el que se muestran los controles de usuario que están instalados en la granja a la que hemos accedido. Este menú se crea de manera dinámica, ya que ninguna granja tiene los mismos componentes que otra, por lo que no podemos crear el menú de manera estática.

En cuanto a la pantalla de visualización de la granja, hay que destacar una cosa muy importante, los controles de usuario que se encuentran posicionados en la pantalla se crean dinámicamente cada vez que se accede a la página, ya que cada granja tiene distintos controles de usuario. Lo importante es, que a la hora de crear estos controles, se crea un objeto genérico (llamado control de usuario genérico). De este modo se ha conseguido que no se tenga que modificar el código de la aplicación cada vez que en las granjas se instale un nuevo control que no existía en el sistema. Así, basta con crear el control de usuario

nuevo y guardarlo en una carpeta de la aplicación (llamada controles de usuario), sin tener que añadir líneas de código nuevas a la aplicación.

Faltan de tratar temas importantes como son la seguridad y el control de accesos, ya que se trata de una aplicación que se va a implantar en instalaciones reales y por tanto un acceso no permitido al sistema puede causar graves perjuicios económicos.

Se ha planteado la posibilidad de utilizar Windows Identity Foundation como plataforma para gestionar la seguridad. Este sistema utiliza certificaciones para asegurarse de que el usuario que está intentando acceder al sistema es seguro. Para ello, la aplicación se configura para que acepte los certificados de una entidad de confianza. Esta entidad será la que decida a quien le concede el certificado o no y con qué características, ya que estos certificados pueden llevar una serie de argumentos que se utilizan para describir al usuario al que pertenecen.

Además, se puede plantear el adaptar la aplicación completa para ser usada en dispositivos móviles. De este modo, el control de las granjas se podrá realizar desde cualquier punto geográfico siempre que se tenga acceso a líneas de internet móvil.

Por último, existe otra posible ampliación, que el sistema envíe sms a los agricultores informando de cualquier tipo de incidencia que esté ocurriendo en sus granjas. Así, los granjeros estarían informados en todo momento acerca del funcionamiento de sus granjas.

Capítulo 5

Bibliografía

[1] Modelo Basado en Componentes. Diseño de sistemas.

Disponible en World Wide Web:

<http://www.mitecnologico.com/Main/ModeloBasadoEnComponentesDise%F1oDeSistemas>

- Introducing Silverlight 4. Ashish Ghoda. En: Apress (ed). 2010.
- Pro Business Applications with Silverlight 4. Chris Anderson. En: Apress (ed). 2010.

GESTIÓN DE SISTEMA DOMÓTICO MEDIANTE PLATAFORMA WEB

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN
PROYECTO FIN DE CARRERA

Alumno: Carlos Ulibarrena Nuin
Tutor: José Javier Astrain Escola

ÍNDICE

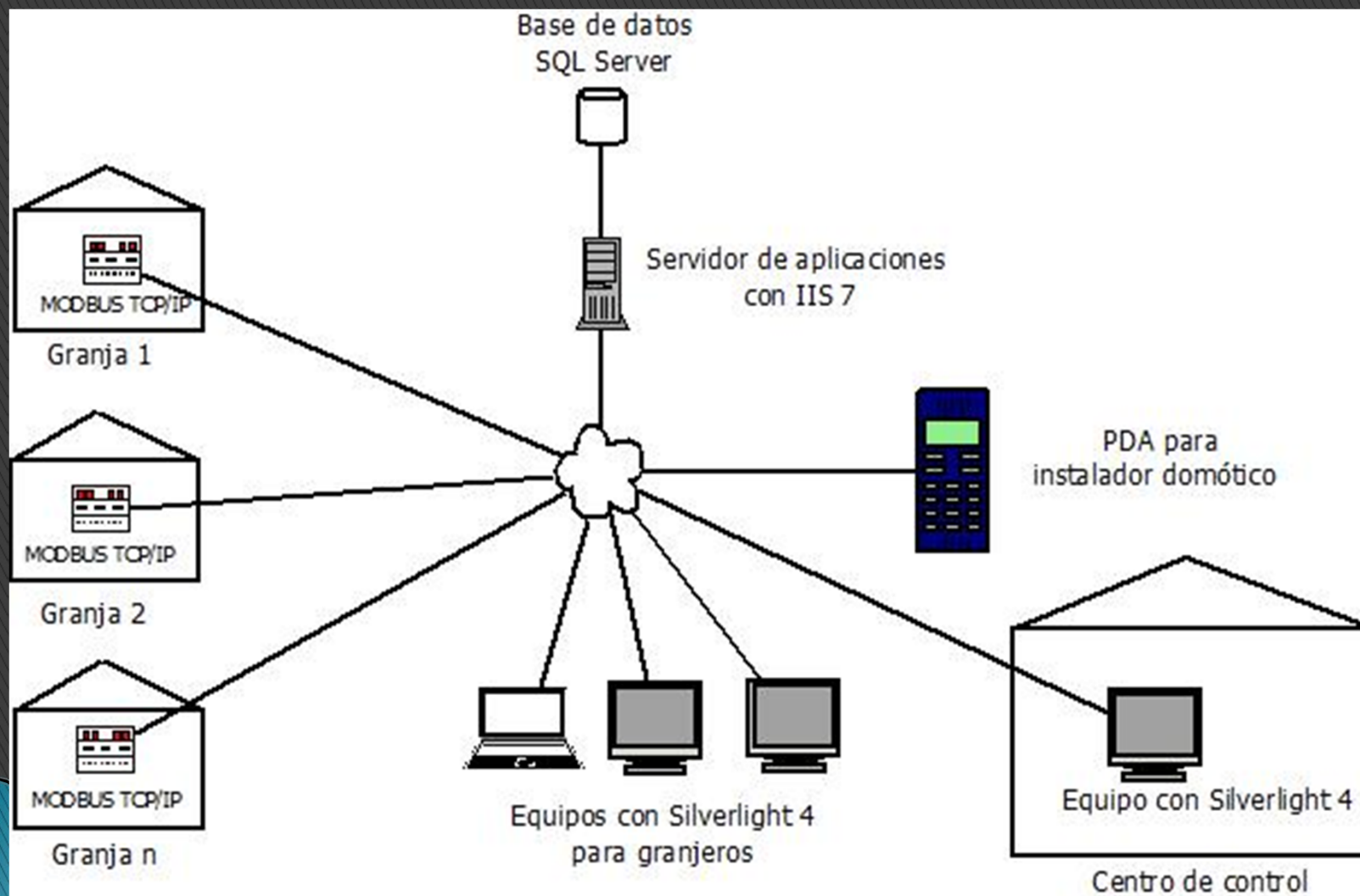
- ▶ Introducción
- ▶ Planificación
- ▶ Metodología
- ▶ Análisis del sistema
- ▶ Diseño del sistema
- ▶ Conclusiones y líneas futuras

ÍNDICE

- ▶ Introducción
- ▶ Planificación
- ▶ Metodología
- ▶ Análisis del sistema
- ▶ Diseño del sistema
- ▶ Conclusiones y líneas futuras

Introducción

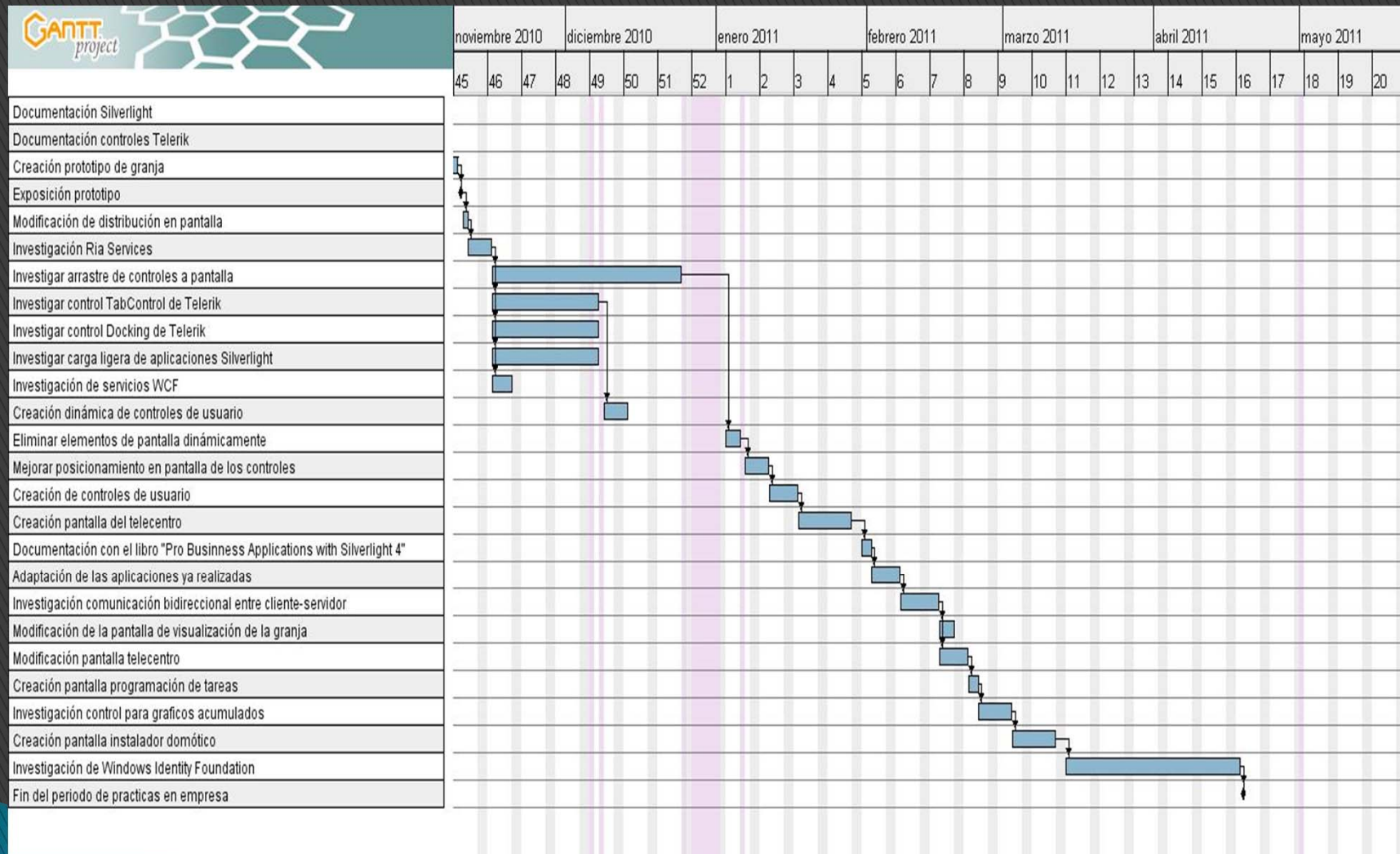
Esquema del sistema



ÍNDICE

- ▶ Introducción
- ▶ Planificación
- ▶ Metodología
- ▶ Análisis del sistema
- ▶ Diseño del sistema
- ▶ Conclusiones y líneas futuras

Planificación

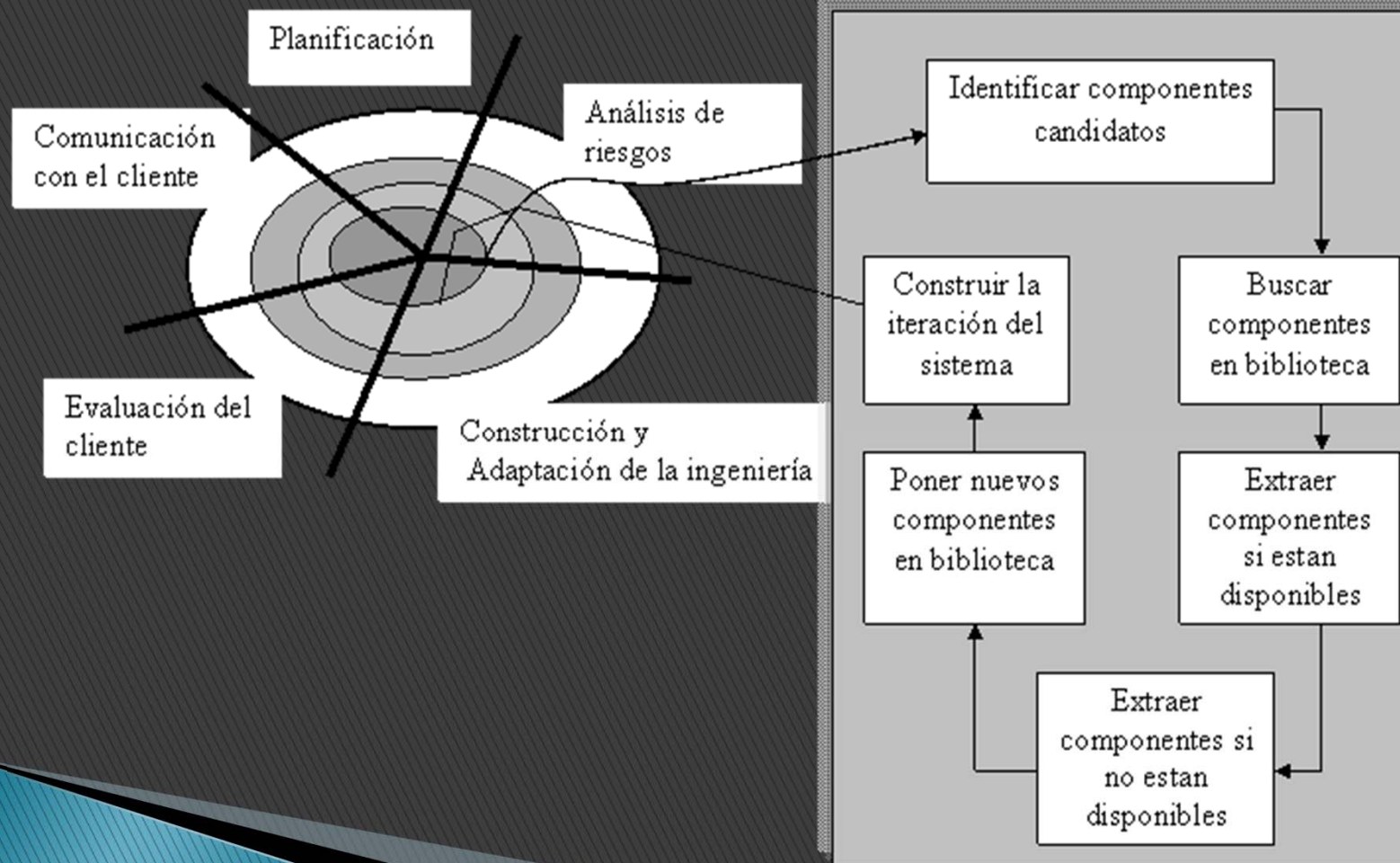


ÍNDICE

- ▶ Introducción
- ▶ Planificación
- ▶ Metodología
- ▶ Análisis del sistema
- ▶ Diseño del sistema
- ▶ Conclusiones y líneas futuras

Metodología

► Modelo de proceso basado en componentes



ÍNDICE

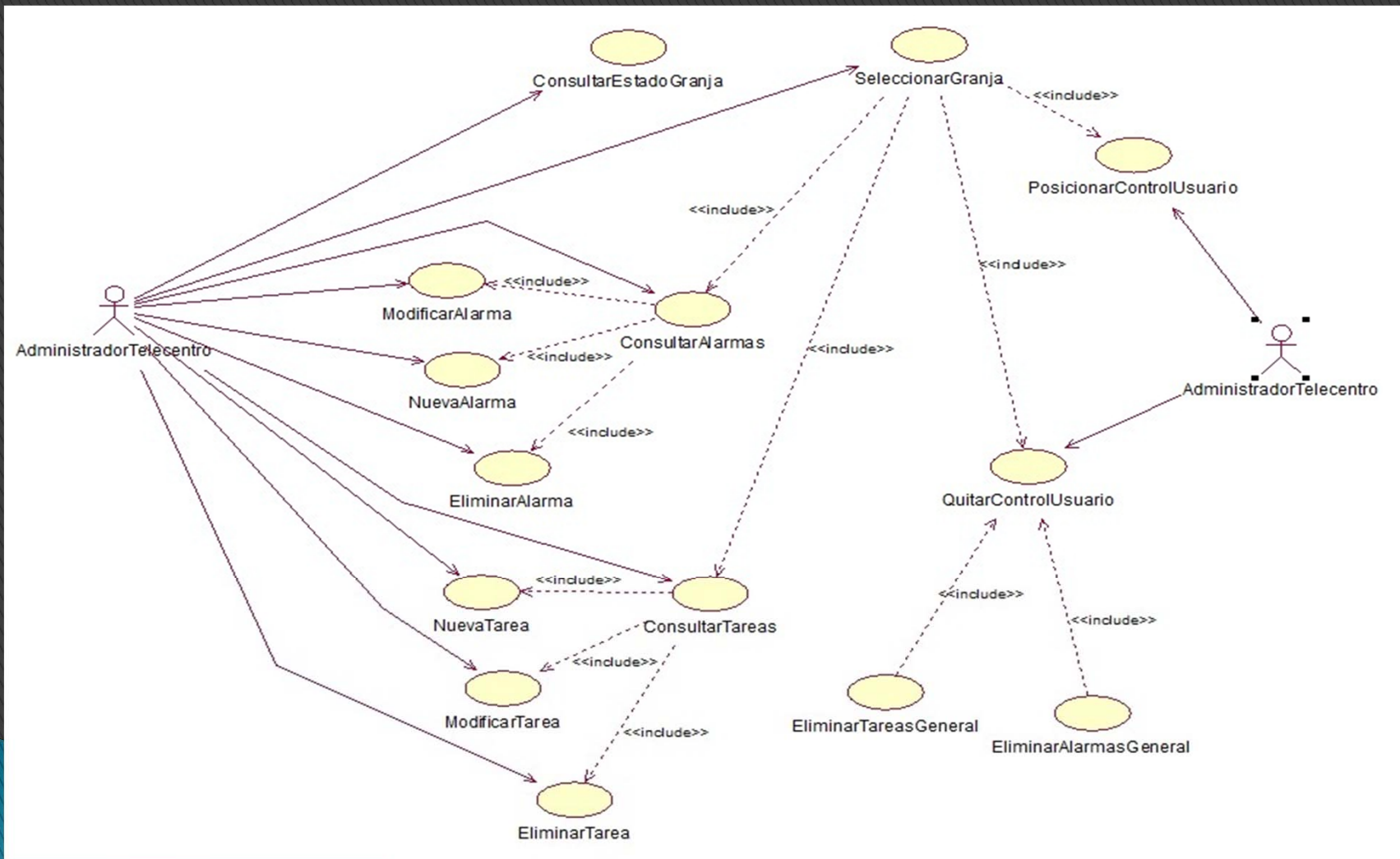
- ▶ Introducción
- ▶ Planificación
- ▶ Metodología
- ▶ **Análisis del sistema**
- ▶ Diseño del sistema
- ▶ Conclusiones y líneas futuras

Análisis del sistema

- ▶ Usuarios del sistema: Granjero, Instalador y Administrador Telecentro.
- ▶ Dividimos los casos de uso según los usuarios que se han identificado.

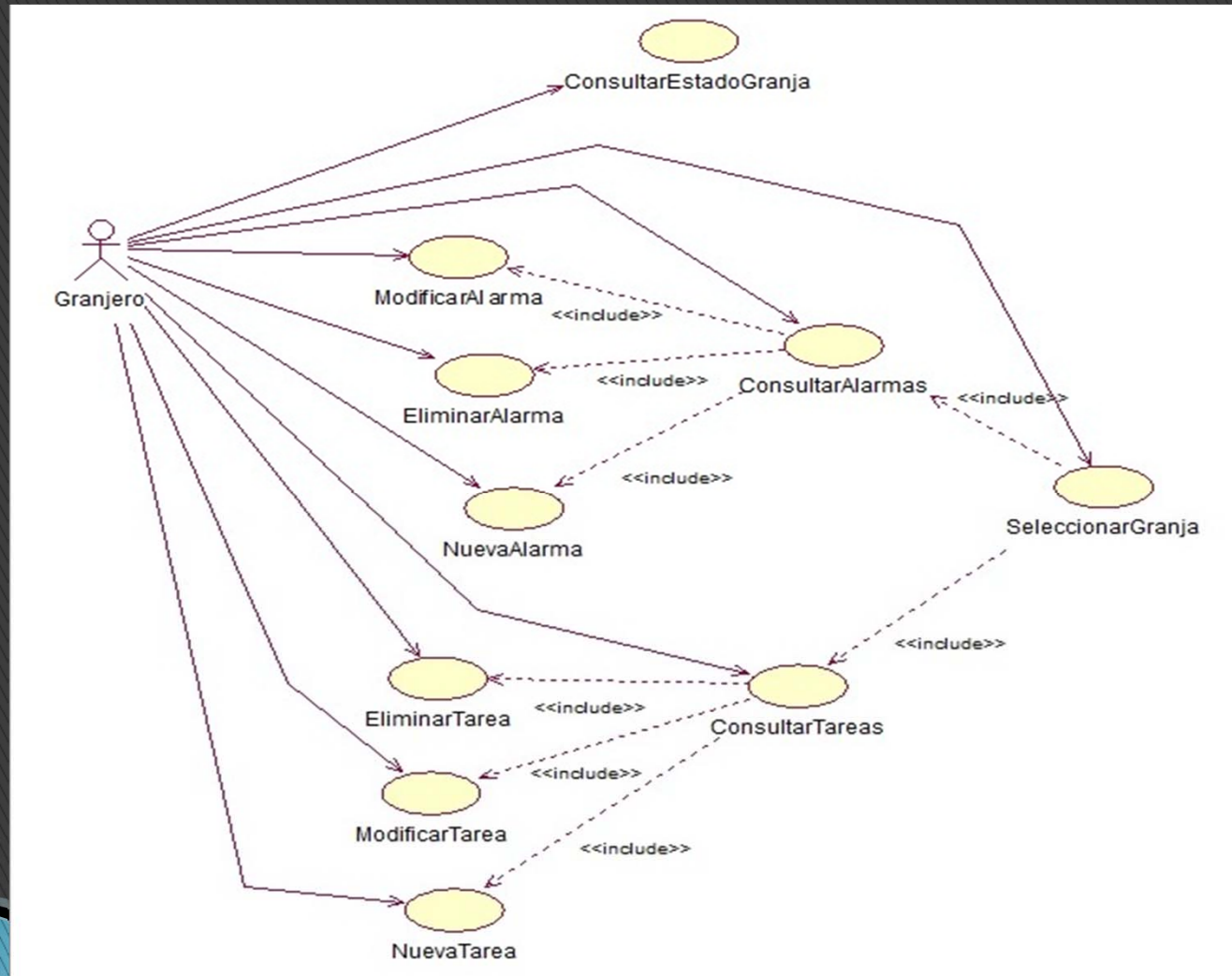
Análisis del sistema

► Casos de uso del usuario AdministradorTelecentro



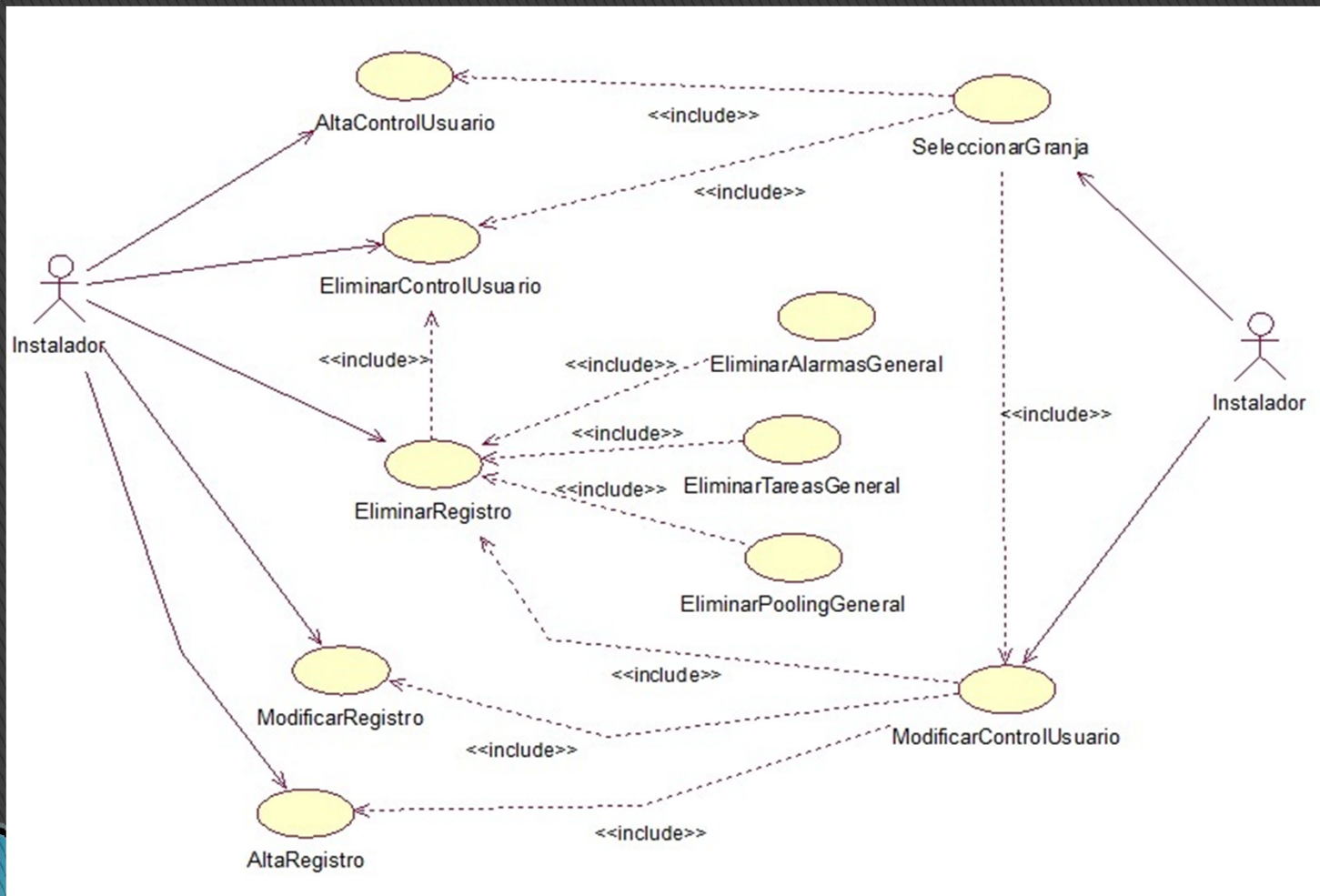
Análisis del sistema

► Casos de uso del usuario Granjero



Análisis del sistema

► Casos de uso del usuario Instalador



ÍNDICE

- ▶ Introducción
- ▶ Planificación
- ▶ Metodología
- ▶ Análisis del sistema
- ▶ Diseño del sistema
- ▶ Conclusiones y líneas futuras

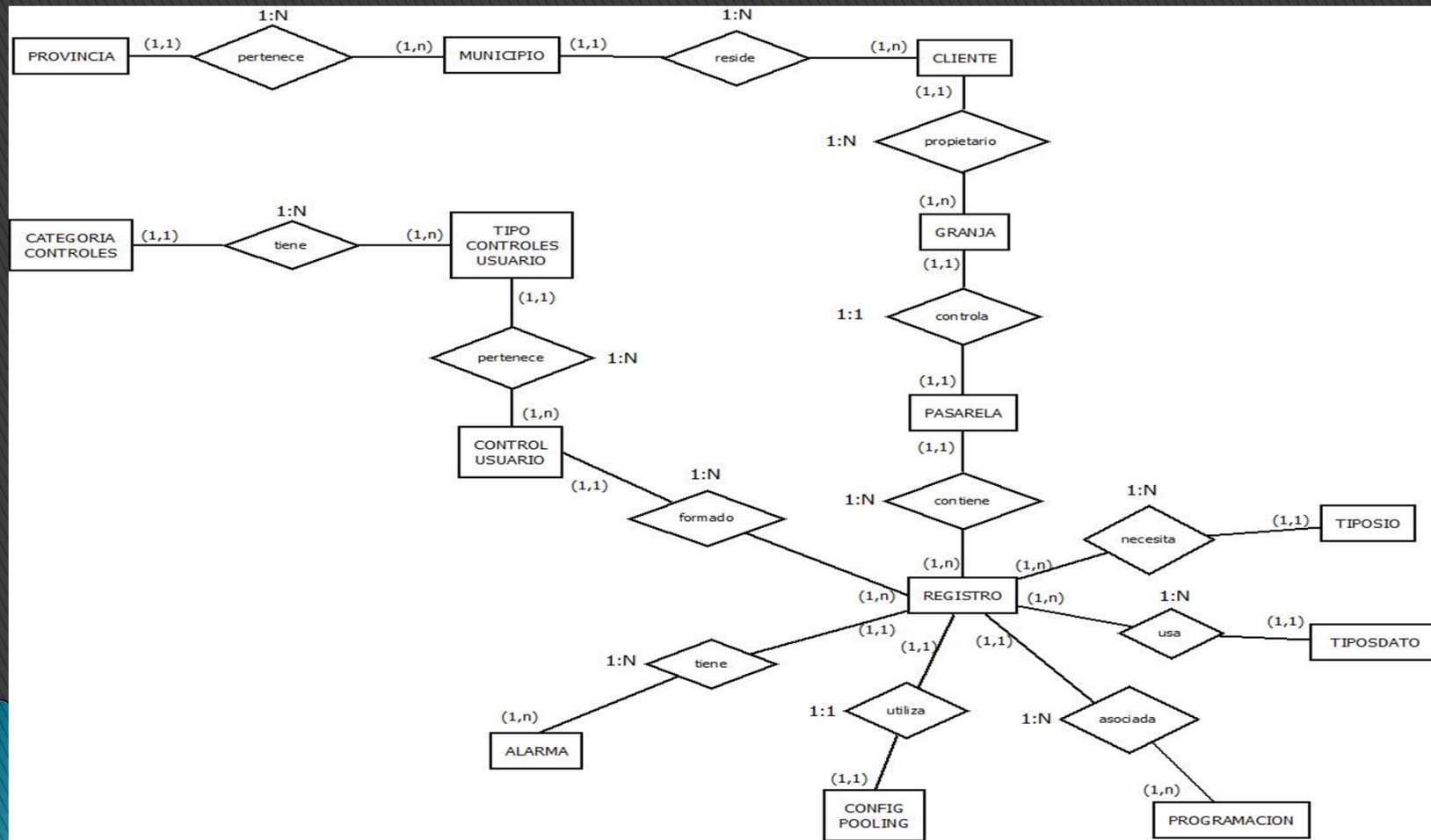
Diseño del sistema

- ▶ Diseño en tres capas: capa de gestión de datos, capa de dominio y capa de presentación.
- ▶ Facilidad para mantener y se favorece la reutilización.

Diseño del sistema

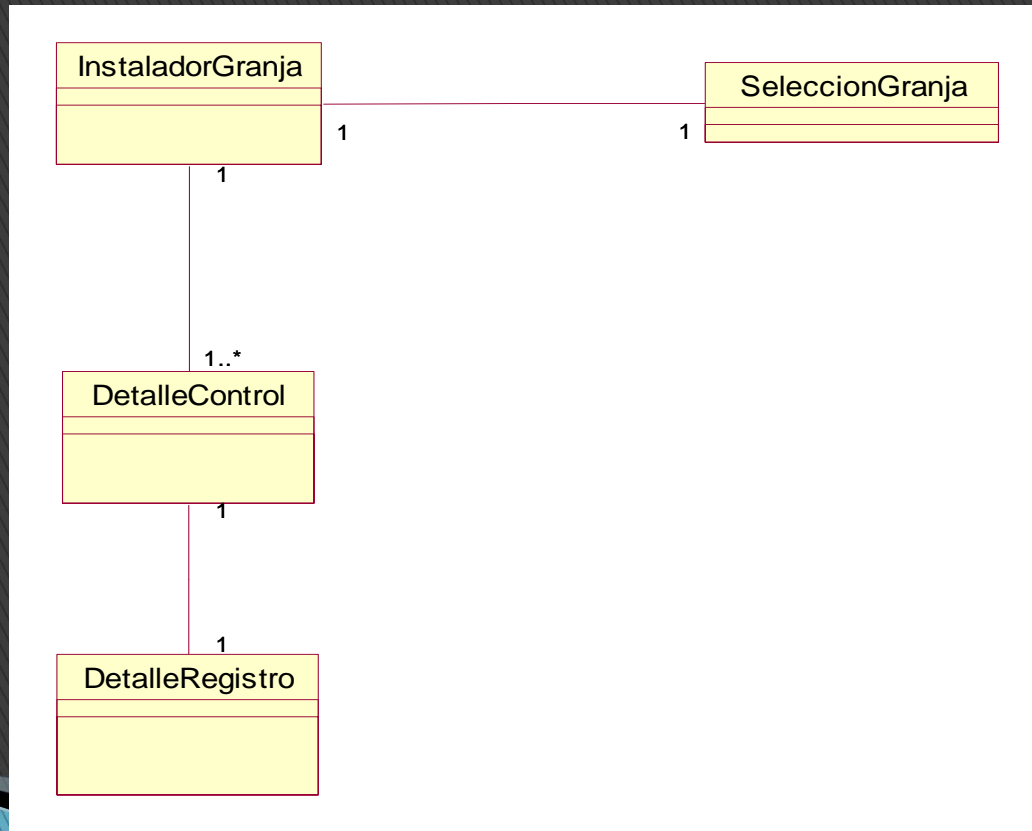
► Capa de Gestión de Datos

◦ Diagrama Entidad-Relación



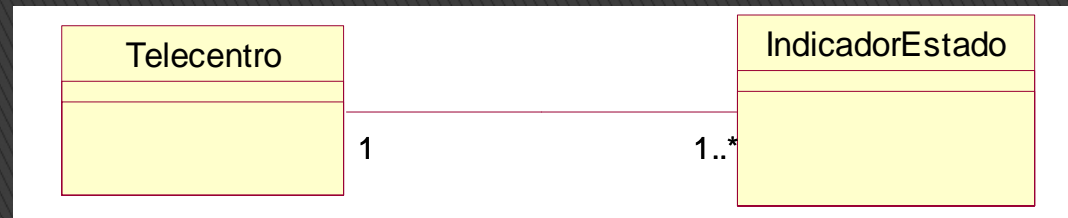
Diseño del sistema

- ▶ Capa de Dominio
 - Módulo InstaladorGranja

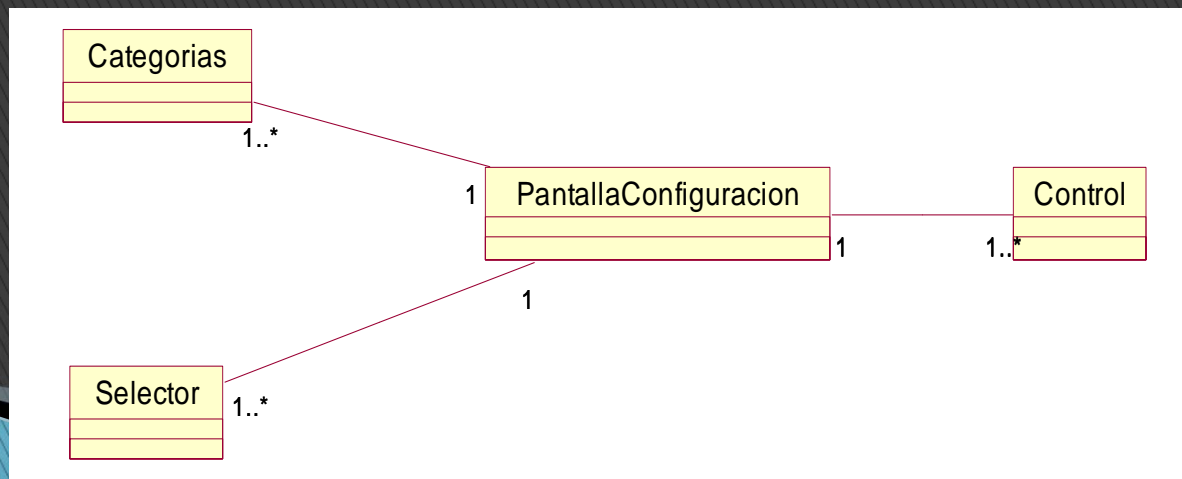


Diseño del sistema

- ▶ Capa de Dominio
 - Módulo Telecentro



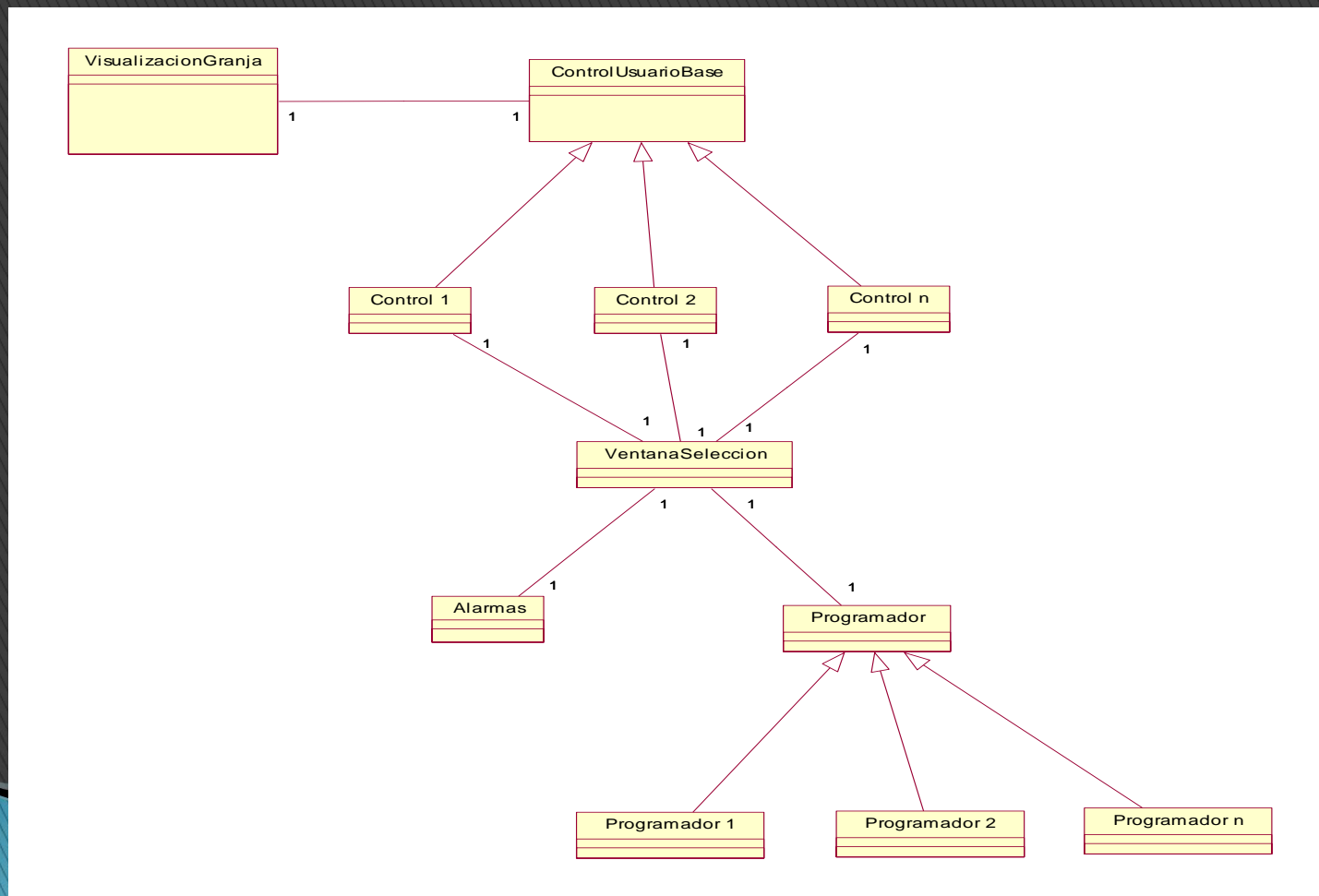
- Módulo PantallaConfiguracion



Diseño del sistema

► Capa de Dominio

◦ Módulo VisualizacionGranja



Diseño del sistema

- ▶ Capa de Presentación
 - Módulo Instalador – DetalleRegistro

Datos del Registro

Código:

Nombre:

Descripción:

Dirección:

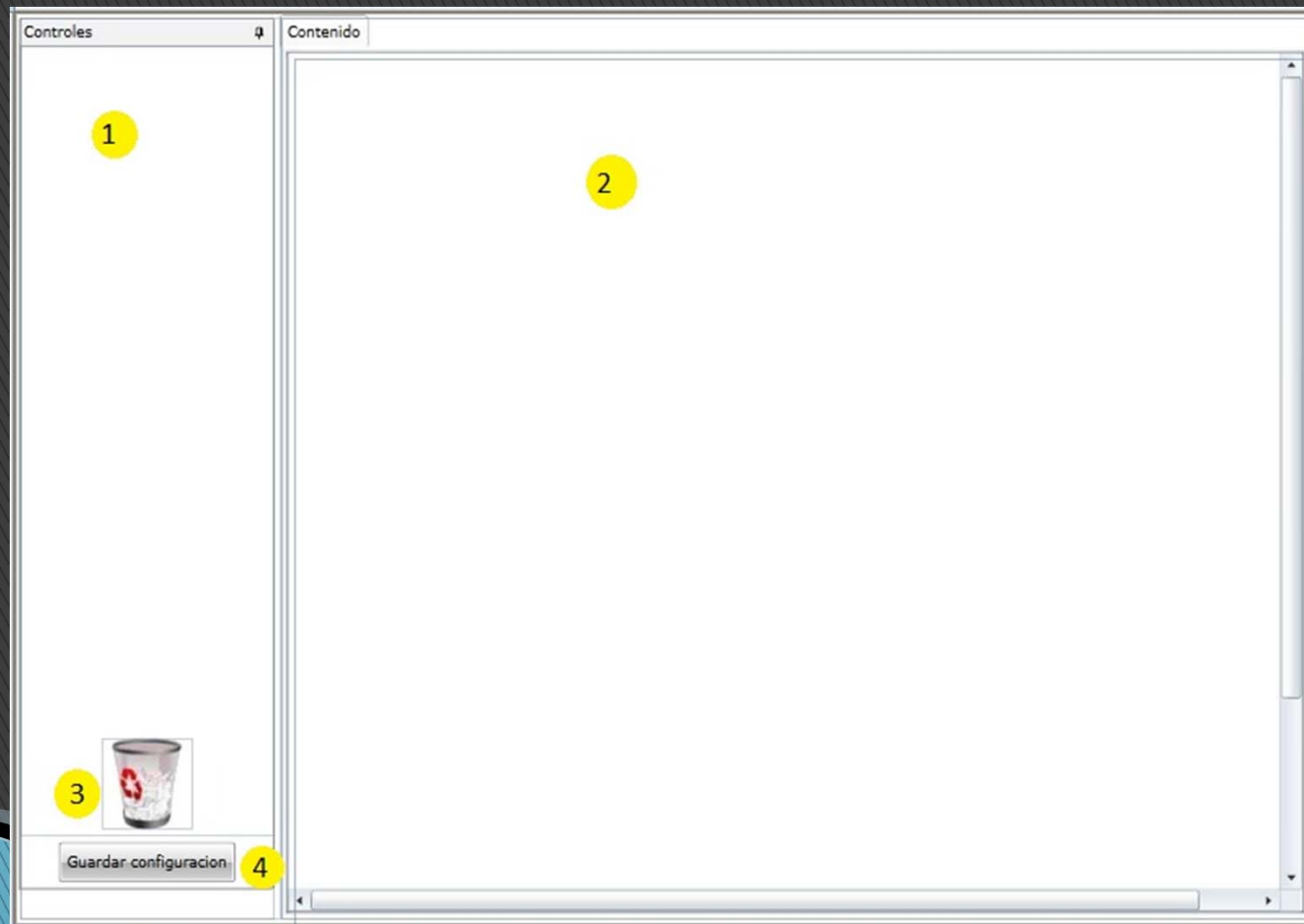
Tipo IO: Tipo Dato:

Valor Mínimo: Valor Máximo: Tipo Dato Analógico:

Refresco Activo: ☐ Intervalo de refresco (minutos):

Diseño del sistema

- ▶ Capa de Presentación
 - Módulo PantallaConfiguracion – PantallaPrincipal



Diseño del sistema

- ▶ Capa de Presentación
 - Módulo PantallaConfiguracion – ControlUsuario



Diseño del sistema

- ▶ Capa de Presentación
 - Módulo Telecentro- PantallaPrincipal



ÍNDICE

- ▶ Introducción
- ▶ Planificación
- ▶ Metodología
- ▶ Análisis del sistema
- ▶ Diseño del sistema
- ▶ Conclusiones y líneas futuras

Conclusiones y líneas futuras

- ▶ Trabajo en empresa e integración.
- ▶ Utilización de nuevas tecnologías de programación.
- ▶ Sistema Drag&Drop y creación dinámica de controles.
- ▶ En un futuro: seguridad y control de accesos, aplicación completa para móviles, avisos vía sms.

Agradecimientos

- ▶ Tribunal
- ▶ EGA Informática
- ▶ José Javier Astrain Escola

¿ Preguntas ?